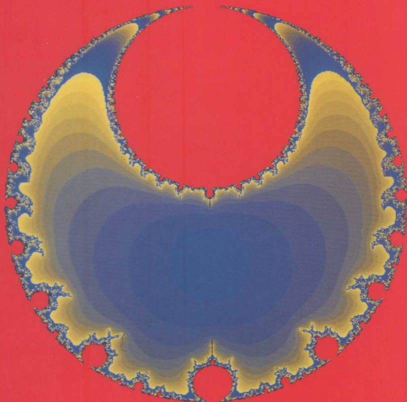
Hans Lauwerier

Een wereld van FRACTALS



ARAMITH UITGEVERS

B

EEN WERELD VAN FRACTALS

HANS LAUWERIER

Een wereld van fractals



Aramith Uitgevers Bloemendaal

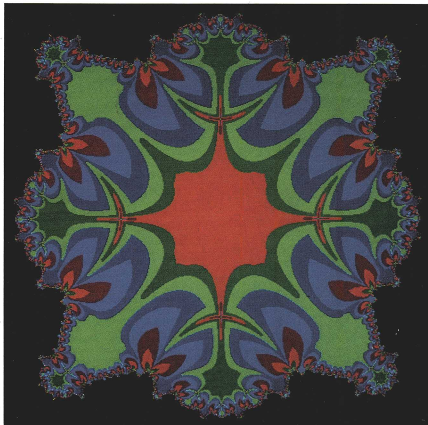
CIP-gegevens Koninklijke Bibiotheek, Den Haag

Lauwerier, Hans

Een wereld van fractals. Hans Lauwerier
Bloemendaal, Aramith
Met bibliografie
ISBN 90 6834 076 X
SISO 527.5 UDC 519.67:681.324 NUGI 819
Trefwoord: fractals

Omslagontwerp: Hard Werken Design
Typografische vormgeving: Marjo Starink/Studio Cursief
Zetwerk: Zetcentrale Meppel bv
Druk: Ten Brink Meppel

INHOUD



Een Julia-tegel bepaald door $z^8 + z^4 - \frac{1}{2}$

Fractals, eens wiskundige curiositeiten, hebben nu de wereld veroverd. Mede dankzij de moderne computertechnologie staan ze in de wetenschap centraal bij computerexperimenten. De toepassingen strekken zich uit over vele terreinen. Een overvloed aan wetenschappelijk materiaal is beschikbaar, zij het meestal in een voor niet-specialisten weinig toegankelijke vorm. In dit boek hebben we ons tot taak gesteld de belangstellende lezer daarin te betrekken met theorie en praktijk. De ideale lezer dient liefst te beschikken over enige wiskundige kennis en over een moderne, snelle computer met de EGA- of VGA-modus, zo mogelijk met zestien kleuren. Wanneer de computer ook nog uitgerust is met een mathematische coprocessor, is het helemaal ideaal. Hij is dan in staat een aantal wetenschappelijke experimenten na te volgen, en een groot aantal fractals op het beeldscherm te toveren. Naarmate men meer wiskundige kennis in huis heeft, kan men ook meer inzicht in de wereld van de fractals verwerven. Men kan dan bijvoorbeeld zelf fractals ontwerpen die aan bepaalde van tevoren gegeven specificaties voldoen, of men kan voorspellen wat het effect is van wijzigingen in een bepaalde formule. Wie op min of meer artistieke wijze zelf Julia sets en Mandelbrot-details wil ontwerpen, zal veel nut kunnen hebben van de kennis van de rekenwijze van de complexe getallen. Die techniek is eigenlijk vrij eenvoudig, maar wordt op school meestal niet behandeld.
De hoofdstukken over de Julia sets en de Mandelbrot set vormen een van de hoogtepunten van dit boek. Weliswaar bevatten deze hoofdstukken relatief veel wiskundige details, maar de tekst is zo geschreven, dat al te technische passages zonder bezwaar overgeslagen kunnen worden. Een ander hoogtepunt is het hoofdstuk over de ruimtelijke fractals. We laten daarin onder andere zien hoe van de bekende Pythagoras-boom ruimtelijke varianten kunnen worden gemaakt, en hoe een zogenaamd fractallandschap, compleet met bergen en zee, geconstrueerd kan worden.
Vergeleken met het vorige boek, *Fractals*, van dezelfde auteur weerspiegelt dit boek de geweldige ontwikkelingen op het gebied van fractals. Wat ooit bedoeld was als een mogelijke nieuwe, uitgebreide druk is nu uitgegroeid tot een volledig nieuw en zelfstandig werk. Daarnaast is *Fractals* nog steeds goed bruikbaar als algemene inleiding.
Dit boek gaat over fractals, een wiskundig onderwerp. Daarover praten kan alleen maar goed, door gebruik te maken van enige wiskundige begrippen en formules. De schrijver heeft ernaar gestreefd het juiste midden te kiezen tussen een wiskundige verhandeling en een vlot, populair-weten-

schappelijk verhaal. Voor wiskundigen missen sommige passages misschien de hun vertrouwde precisie, maar zij zullen weinig moeite hebben de tekst hier en daar waar nodig aan te scherpen. Zij worden geacht de goede verstaanders te zijn. Termen die tot het jargon van de wiskundige behoren, worden vaak of apart uitgelegd, of gaan vergezeld van synoniemen. Wanneer hier en daar ongeveer hetzelfde wordt gezegd, is dit gedaan ten gerieve van de minder deskundige lezer. Voor hem geldt het advies, gewoon doorlezen en proberen te wennen aan de aanvankelijk vreemde termen, zoiets als het lezen van een Engelstalige detective zonder woordenboek.

Bij het schrijven van dit boek is dankbaar gebruik gemaakt van de in de bibliografie vermelde boeken en artikelen. Bij de programma's en de illustraties is gebruik gemaakt van ideeën van deskundige mede-computergebruikers. In het bijzonder vermelden we A. G. Kuiper, die zich een ware ontdekkingsreiziger in het land van Mandelbrot toonde. Aan hem danken we de heel fraaie illustraties op de pagina's 45, 81 en 144. De rest van de illustraties is van de hand van de auteur.

Ook dit boek noodt tot zelf experimenteren. Wie beschikt over een moderne MS-DOS computer zal de in Basic geschreven programma's zonder meer in een gecompileerde vorm, Turbo Basic, Power Basic of Quick Basic bijvoorbeeld, kunnen gebruiken. Minder kan natuurlijk ook, maar dan zal men misschien lang moeten wachten en zal het nodig kunnen zijn in de programma's kleine aanpassingen aan te brengen.

Voor de illustraties is meestal gebruik gemaakt van een configuratie bestaande uit een Compaq 386/20e met een Eizo 9070S beeldscherm en een HP Laserjet II. De zwart-witte illustraties zijn tot stand gekomen door de Laserjet als HP-plotter te gebruiken. De originele resolutie is daarbij 300 pixels per inch. De kleurenplaatjes zijn verkregen via rechtstreekse fotografie van het beeldscherm met behulp van standaardapparatuur.

De achter in het boek opgenomen programma's hebben een dubbel doel. Aan de ene kant dienen ze als technische aanvulling op de tekst, maar daarnaast stellen ze de achter de computer zittende lezer in staat ook zelf zulke mooie fractalplaatjes te maken. Veel programma's bevatten variabele elementen, getallen die door de gebruiker naar eigen inzicht en fantasie veranderd kunnen worden. Aldus kan reeds een enkel programma toegang geven tot een wereld van fractals.

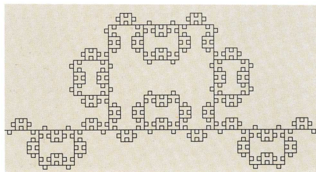
Verkenning

Menige lezer zal wel eens met het begrip 'fractal' kennisgemaakt hebben. Wie thuis een computer heeft en de kunst verstaat daarmee fraaie plaatjes zichtbaar te maken, heeft ze misschien al vele malen op het beeldscherm gehad. Er is trouwens nogal wat zogenaamde 'public domain software' in omloop met behulp waarvan men eindeloos kan experimenteren. Termen als 'Julia sets' en de 'Mandelbrot set' hebben een vertrouwde klank gekregen, ook al begrijpen we er misschien maar weinig van. In natuurwetenschappelijke kring worden fractals steeds meer gezien als een nieuw meetkundig begrip, als objecten waarmee men zulke uiteenlopende zaken als het wortelgestel van een plant, het bloedvatenstelsel van de lever, de geulen van de Waddenzee, het kraterlandschap van de maan, de structuur van een metaaloppervlak, de vorm van koralen en sponzen kan beschrijven.
In dit eerste hoofdstuk willen we de voornaamste wetenswaardigheden omtrent fractals op een rijtje zetten. De meeste zaken die hier aan de orde komen, zullen later in dit boek uitvoeriger worden behandeld. Om dieper op de theorie in te kunnen gaan is wel enige kennis van wiskunde nodig. Omdat dit boek bedoeld is voor een algemeen publiek, hebben we ons daarbij de uiterste beperkingen opgelegd. Eigenlijk is alleen wat kennis van de (vlakke) coördinatenmeetkunde nodig en wat vertrouwdheid met functies als de sinus, de cosinus, de logaritme en de exponentiële functie, het basismateriaal voor het grafisch werken met de computer.
Voor meer technische zaken moeten we de lezer verwijzen naar wetenschappelijke literatuur welke ruimschoots voorhanden is, zij het dat nog slechts een klein deel in boekvorm verkrijgbaar is.

Wat is een fractal?

Een fractal is een ingewikkelde meetkundige figuur, waarin men een zekere mate van zelfgelijkvormigheid kan aantreffen.
Geen erg wetenschappelijke definitie, maar veel beter is er niet. Met zelfgelijkvormigheid bedoelen we dat een willekeurig klein deel van de fractal alle elementen van het geheel bevat. Zouden we een fractal onder een microscoop leggen, dan zouden we onafhankelijk van de gebruikte vergroting toch ongeveer hetzelfde plaatje zien, maar ook dat moeten we niet al te letterlijk nemen. Het is zoiets als de definitie van 'meetkunde'. Toen een wiskundige daarnaar gevraagd werd, was zijn antwoord: 'Alles wat een gezaghebbend wiskundige daaronder wil verstaan.' Voor fractals is Mandelbrot onbetwistbaar 'de grote roerganger'. Oorspronkelijk baseerde

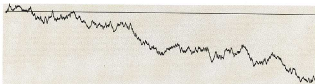
hij de definitie op het begrip van een dimensie met een niet-geheeltallige waarde. Het is gebruikelijk om aan punt, lijn en vlak de dimensies 0, 1 en 2 toe te kennen. Er bestaan in de wiskunde echter ook andere definities, waarbij de dimensie ook tussengelegen waarden kan aannemen, zogenaamde gebroken waarden. De term fractal is dan ook afgeleid van 'fractal dimension' of 'gebroken dimensie'. In het tweede hoofdstuk gaan we er dieper op in. Het zal dan blijken dat de fractal(e) dimensie van de in figuur 1.1 afgebeelde lijn gelijk is aan log 6/log 3 = 1.58. Het kan echter heel goed gebeuren dat een fractal toch de dimensie 1, als een cirkel, of 2, als een gevulde cirkelschijf, heeft. Dat was dan ook de reden voor Mandelbrot om met een nieuwe definitie te komen, waarin de zelfgelijkvormigheid op de voorgrond staat.
Zelfgelijkvormigheid kan aanwezig zijn in strikt meetkundige zin, zodanig dat elk detail, hoe klein ook, alle kenmerken van het geheel bevat. Een goed voorbeeld is de in figuur 1.1 afgebeelde kromme van Mandelbrot en Given, een variant van de bekende Von Koch-lijn.



Figuur 1.1 De lijn van Mandelbrot en Given

Dergelijke fractals, waaraan een meetkundig principe ten grondslag ligt, zullen we betitelen als meetkundige fractals. Er kan echter ook sprake zijn van meer toevallige overeenkomsten, waarbij details niet onderling gelijkvormig zijn, maar alleen op elkaar lijken. We kunnen dan alleen over een soort gemiddelde zelfgelijkvormigheid spreken. Wie naar een wolkenlucht kijkt, schapenwolkjes bijvoorbeeld, begrijpt wel wat we bedoelen. Wie vanuit een vliegtuig hoog boven de oceaan naar beneden kijkt, ziet rimpelingen op een wateroppervlak alsof hij zich vlak boven een door een zuchtje wind beroerd vijvertje bevindt. In werkelijkheid ziet hij slechts de grootste golven. Het is het macroscopische equivalent van wat de bioloog Robert Brown in zijn microscoop waarnam en wat nu Brownse beweging heet. In figuur 1.2 hebben we het experiment van Brown in sterk vereenvoudigde vorm met behulp van de computer gesimuleerd. Een deeltje kan zich alleen verticaal bewegen. Bij elke tik van een imaginaire klok doet het deeltje of een stapje omhoog, of omlaag, waarbij alle stapjes even groot, of liever even klein zijn. Het is als het telkens werpen van kruis of munt. Werpen we munt, dan verdienen we een cent, maar bij kruis verlie-

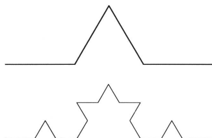
Figuur 1.2 De lijn van Brown



zen we er een. Figuur 1.2, vervaardigd met behulp van het programma BROWNL, kunnen we dan interpreteren als de grafiek van ons saldo in de loop van de tijd.
Dergelijke fractals, waarvan de structuur berust op de wetten van de waarschijnlijkheidsrekening (stochastiek), zullen we stochastische fractals noemen. In de hoofdstukken 7 en 10 komen we er uitvoerig op terug.

Een ideale fractal bestaat niet

We beperken ons in dit boek hoofdzakelijk tot meetkundige figuren in een plat vlak en dan zijn de rechte lijn, of een lijnstuk, en de cirkel de eenvoudigste basispatronen. Men zou ze zelfs kunnen opvatten als fractals, omdat ze zelfgelijkvormig zijn, maar ze zijn te eenvoudig om er dat trefwoord aan te verbinden. Men dient overigens te bedenken dat rechte lijn en cirkel idealisaties zijn van figuren die met potlood en pen op een of ander medium getrokken worden. Een wiskundige lijn heeft geen dikte, is oneindig dun zegt men wel. Zo'n lijn zou dus onzichtbaar moeten zijn. Een ideale cirkel bestaat dus alleen in onze gedachtenwereld. In de praktijk moet men genoegen nemen met benaderingen, moet men zijn toevlucht nemen tot dikke krijtstrepen of tot inktlijnen getrokken met een passer. Dat zelfde geldt ook voor fractals. De in figuur 1.1 afgebeelde kromme is natuurlijk maar een benadering. De ideale vorm van deze fractal kunnen we alleen in onze gedachten vormen. De echte fractal is zelfgelijkvormig in alle details, ziet er eender uit bij welke vergroting ook. Maar wanneer we een klein detail van figuur 1.1 vergroten, zien we al gauw niet meer dan een recht lijnstuk. In de praktijk maken we ons daar maar niet te druk over. In de regel zorgen we ervoor dat de illustraties, hetzij op papier, hetzij op het beeldscherm van de computer in overeenstemming met het beschikbare oplossend vermogen, voldoende gedetailleerd zijn.
Om fractals te begrijpen, volgen we een soort natuurmethode. De beste methode om een taal te leren is proberen zich verstaanbaar te maken door het zelf vormen van woorden en zinnen. De nodige kennis doen we op door goed naar anderen te luisteren. En passant nemen we dan ook een stukje grammatica mee. Zo ligt in dit boek het accent van het zelf construeren van fractals eerst op het navolgen van de klassieke voorbeelden en dan op het aanbrengen van eigen variaties. We gaan dus fractals maken, en daartoe is een aantal methoden of principes beschikbaar. Die zijn zo belangrijk en zo interessant, dat ze aparte hoofdstukken waard zijn. In grote lijnen kennen we het principe van de *recursie*, de *contractie*-methode en de *dynamische systemen*. Over elk van die onderwerpen zullen we hieronder iets zeggen.



Figuur 1.3 De recursieve constructie van de Von Koch-lijn

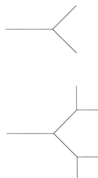
De recursie

Recursief betekent letterlijk 'teruglopend' en wil zeggen dat een bepaalde redenering, een meetkundige constructie, almaar herhaald wordt. Voor bijvoorbeeld de lijn van Von Koch, afgebeeld in figuur 2.2, kan de methode op eenvoudige wijze toegelicht worden. Alles berust op een enkele meetkundige handeling:

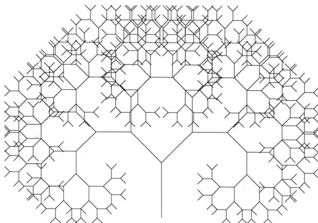
'Van een recht lijnstuk wordt het middelste derde gedeelte vervangen door de opstaande zijden van een gelijkzijdige driehoek.'

Die bewerking is afgebeeld in figuur 1.3. De rechte lijn wordt dus vervangen door een gebroken lijn van vier stukjes, waarbij de totale lengte 4/3 van die van het origineel is. Die bewerking wordt oneindigvoudig herhaald, ook weer een zich in onze gedachten afspelend proces. De bij elke stap ontstane kleinere lijntjes worden op hun beurt ook weer aan dezelfde operatie onderworpen. Ten slotte, na oneindig veel handelingen, bevat de kronkellijn geen rechte stukjes meer. Het is de bekende lijn van Von Koch geworden, historisch gesproken een van de oudste fractals. Inmiddels is de lengte wel oneindig geworden. Het is duidelijk dat we in de praktijk ergens moeten ophouden, en dat hangt natuurlijk af van de gebruikte hulpmiddelen.
Vaak leidt de recursieve constructie tot een boomstructuur. Als voorbeeld beschouwen we de in figuur 1.4 geschetste vorming van kleinere zijtakken aan een reeds gevormde tak. We nemen aan dat de zijtakken hoeken van 45 graden maken met de hoofdtak en dat ze verkort zijn in de verhouding van $1/\sqrt{2}$.
Beginnen we met een enkel recht lijnstuk en passen we die constructie een flink aantal keren toe, dan ontstaat met behulp van PYT3BT figuur 1.5, een zogenaamde kale boom van Pythagoras, ook besproken in *Fractals*. Het is niet moeilijk te bewijzen dat de fractaldimensie van deze fractal juist 2 is, de dimensie van een gevuld vlakdeel. Dat wil zoiets zeggen dat naarmate de takjes van die boom steeds kleiner worden, ze met elkaar steeds vlakvullender worden. Volgens de oorspronkelijke definitie van Mandelbrot zouden we die fraaie figuur geen fractal mogen noemen, en dat zou toch wel jammer zijn.
De boomstructuren zijn zo belangrijk, dat hoofdstuk 5 er geheel aan gewijd is. In het boek *Fractals* kan men een groot aantal voorbeelden vinden van het gebruik van de recursieve methode. Men kan zich voorstellen dat computerprogrammeurs ook in hun programma's met succes van recursieve technieken gebruik kunnen maken. Een nadeel is overigens wel, dat sommige veelgebruikte programmeertalen geen recursieve opdrachten, opdrachten die zichzelf kunnen aanroepen, kennen. Bovendien leiden re-

Figuur 1.4 De recursieve vorming van de takken van een boom



Figuur 1.5 De kale boom van Pythagoras

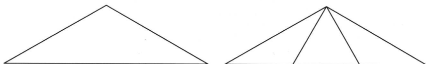


cursieve programma's nogal eens tot een overbelasting van de geheugencapaciteit.
In hoofdstuk 8 schenken we aandacht aan een speciale en heel interessante recursieve techniek. Het betreft met name de zogenaamde Lindenmayer-systemen en hun toepassing in de biologie.

Het contractie-principe

Hierbij staat de zelfgelijkvormigheid voorop. In de eenvoudigste situatie zijn er twee gelijkvormigheidstransformaties L en R, die de fractal als geheel transformeren in een deel van zichzelf. Voor de kromme van Von-Koch zijn ze aangegeven in figuur 1.6. De fractal is bevat in driehoek ABC. De gelijkvormigheidstransformatie L voert de driehoek over in ACD en op overeenkomstige wijze is de driehoek ABC volgens de transformatie R gelijkvormig met CBE. De verkleiningsfactor, de factor waarmee een lijnstuk verkleind wordt door de gelijkvormigheidstransformatie, is gelijk aan de verhouding AC/AB en dus gelijk aan $1/\sqrt{3}$. De oppervlakte van de driehoeken is dienovereenkomstig verkleind met het kwadraat van die factor, $1/3$ dus, iets wat we ook rechtstreeks uit de figuur kunnen aflezen. Een transformatie waarbij lijnstukken verkleind worden heet een *contractie*, en we kunnen constateren dat de fractal zowel door L als door R in een deel van zichzelf overgevoerd wordt.
De twee contracties L en R geven aanleiding tot een hele familie van ge-

Figuur 1.6 De twee contracties van de lijn van Von Koch



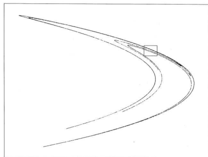
lijkvormigheidstransformaties en elk van die transformaties voert de hele Von Koch-lijn over in een deel van zichzelf. Het is interessant dat L en R de Von Koch-fractal geheel vastleggen. Ook al zou men die fractal nog niet kennen, dan zouden alleen L en R hem volledig definiëren. De kromme van Von Koch is met andere woorden de kleinste meetkundige figuur die met zichzelf gelijkvormig is ten opzichte van L, R en alle combinaties daarvan. Hoe de fractal daadwerkelijk geconstrueerd kan worden komt in de volgende paragrafen aan de orde. In hoofdstuk 3 bespreken we de verschillende soorten contracties in samenhang met het meer algemene begrip transformatie.

Dynamische systemen

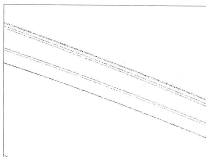
Bij de dynamische methode wordt de fractal opgevat als de limietfiguur van een dynamisch systeem. Een dynamisch systeem is een heel algemeen begrip, een soort gebeuren in de tijd waarbij een baan gevormd wordt. Zo'n baan kan een puntenrij zijn waarbij voor elk punt in het platte vlak een zogenaamde opvolger is gedefinieerd. Uitgaande van een beginpunt P_0 kunnen we dan door herhaling een zogenaamde baan P_0, P_1, P_2, P_3,... vormen waarbij P_1 de opvolger is van P_0, P_2 de opvolger van P_1, enzovoort. In het algemeen bevindt zich ergens een limietverzameling waar de baan heengaat, de zogenaamde aantrekker van het systeem. Soms bestaat zo'n aantrekker slechts uit een enkel punt of een groepje van punten, weinig boeiend dus. Maar vaak is de limietverzameling dusdanig ingewikkeld, dat men van een 'strange attractor' spreekt, een in vakkringen veelgebruikte term die wij vertalen als een *vreemde aantrekker*. In de regel is dat een fractal, al is het vaak moeilijk om dat in alle wiskundige strengheid ook werkelijk te garanderen. Een aantrekker is te vergelijken met een stabiel evenwicht van een mechanisch systeem. Een kleine verstoring van een stabiel evenwicht brengt het systeem naar zijn evenwicht terug. Analoog gaat een baan waarvan het begin zich in de buurt van de aantrekker bevindt ernaar toe. Hebben we het geluk dat het beginpunt zich reeds op de aantrekker bevindt, dan ligt ook de hele baan erop. Het beroemdste voorbeeld is de zogenaamde aantrekker van de Franse astronoom Michel Hénon (1969). Het bewuste dynamische systeem is in coördinatenmeetkunde bepaald door

(1.1)
$x' = y + 1 - ax^2$
$y' = bx$

waarbij $a = 1.4$ en $b = 0.3$ geschikte waarden zijn.
De zogenaamde Hénon-aantrekker is afgebeeld in figuur 1.7.
De limietfiguur, iets als een oneindigvoudig opgevouwen parabool, blijkt niet af te hangen van de gekozen baan. Kiest men een willekeurig beginpunt, dan zijn er in het algemeen maar twee mogelijkheden: of de baan vlucht weg naar het oneindige, of de baan gaat naar de Hénon-aantrekker toe.
In figuur 1.8 laten we nog een paar details zien. Het blijkt dat wat op het



Figuur 1.7 De aantrekker van Hénon



Figuur 1.8 Een detail van de Hénon-aantrekker

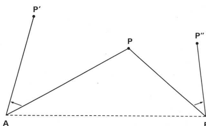
eerste gezicht een enkelvoudige lijn lijkt, bij vergroting een ingewikkelde fijnstructuur vertoont. We zouden in gedachten almaar kleinere fragmenten kunnen vergroten met herhaling van dezelfde detailstructuur. Maar om dat met behulp van de computer uit te voeren, zijn we wel genoodzaakt op zijn minst vele miljoenen baanpunten uit te rekenen. Voor een moderne microcomputer is dat echter geen enkel probleem. Het programma HENATT maakt het mogelijk illustraties als hier afgebeeld *ad libitum* te herhalen.

Bij de dynamische methode wordt een fractal puntsgewijs opgebouwd. Het kan gebeuren dat die punten zich toch weer aaneenrijgen tot een doorlopende lijn, maar het kan ook voorkomen dat de punten los van elkaar blijven zitten. In het laatste geval spreken we van een *stoffractal*, in tegenstelling tot de *lijnfractal* van de Koch-achtige constructie.

In hoofdstuk 4 wordt het onderwerp uitvoeriger behandeld. We wijzen daarbij in het bijzonder op het model van Verhulst en het desbetreffende bifurcatiediagram als voorloper van de in hoofdstuk 13 besproken Mandelbrot set.

Een representatief voorbeeld

Aan de hand van een concreet voorbeeld laten we zien dat bij een zelfde fractal verschillende methoden, principes en technieken van toepassing kunnen zijn. Het gaat om een constructie van punten die men op ruitjespapier gemakkelijk, althans voor de eerste stadia, kan navolgen. Het constructieprincipe is afgebeeld in figuur 1.9. Er zijn twee punten A en B die de centra zijn van contracties. De contractie L heeft A als centrum en transformeert een punt P in een punt P' op een zodanige wijze, dat AP' met AP een hoek van 45 graden maakt in positieve zin en daarbij verkort wordt in de verhouding $1/\sqrt{2}$.



Figuur 1.9 Het constructieprincipe van een fractal

Op analoge wijze heeft de contractie R het centrum B en zijn rotatiehoek en verkortingsverhouding als bij L, maar moet de draaiing nu in negatieve zin genomen worden.
De fractal denken we opgebouwd door als eerste punt het punt C te kiezen dat met AB een rechthoekige, gelijkbenige driehoek maakt. Volgende punten van de fractal worden hieruit afgeleid door telkens in een willekeurige volgorde de transformaties L en R uit te oefenen. De constructie is eigenlijk dezelfde als die voor de kale boom van Pythagoras, het enige verschil is dat we nu punten in plaats van lijnstukjes als transformatieobjecten nemen. Het aantal punten neemt toe volgens de reeks 1, 2, 4, 8, 16,... De totaliteit van alle punten is een fractal die zelfgelijkvormig is ten opzichte van de contracties die uit L en R afgeleid kunnen worden. De constructie van de opeenvolgende generaties punten is als een dynamisch proces. De aantrekker van dat proces is identiek met de reeds gevonden fractal, mits we daaraan de limietpunten hebben toegevoegd, dat wil zeggen als we oneindig ver doorgegaan zouden zijn. De gevormde fractal kunnen we ook opvatten als de limietpunten van de in figuur 1.5 afgebeelde kale boom van Pythagoras, zoiets als de eindpunten van de allerkleinste takjes. Dat laatste moet men natuurlijk niet als een wiskundige uitspraak beschouwen. In de limiet zijn oneindig veel 'kleinste' takjes alle tot een punt ineengeschrompeld. Het merkwaardige feit doet zich nu voor dat al die puntjes onderling verbonden zijn door een continue kronkellijn, die bekend staat als de lijn van Lévy. In de figuren 2.5 en 2.6 zijn daarvan illustraties gegeven.

Julia-fractals

In 1919 beschreef de Franse wiskundige Gaston Julia een bijzonder interessante klasse van fractals, die nu naar hem genoemd zijn. In het Engelse jargon worden ze ook vaak Julia sets genoemd. Daarbij moet men niet aan een sportterm denken, maar dient men te weten dat 'set' de wiskundige term is voor 'verzameling'. Ze geven aanleiding tot buitengewoon fraaie figuren, vooral wanneer de computer ons in staat stelt hun detailstructuur in een rijk kleurengamma te ontrafelen. Ze zijn gedefinieerd door een dynamisch systeem, waarbij de wiskundige structuur bepaald is door een eenvoudige wiskundige functie f, bijvoorbeeld een veelterm of een quotiënt van veeltermen. Die functie komt meetkundig neer op een bepaald soort transformatie van punten in het platte vlak. Voor een passende wiskundige beschrijving dient men enige kennis van complexe getallen te hebben. Veel weet daarvan behoeft men niet te hebben, en wat daarvan beslist nodig is wordt behandeld in appendix 1. De eenvoudigste functie f is de kwadratische. Het overeenkomstige dynamische systeem is dan

(1.2)
$$z' = z^2 + c$$

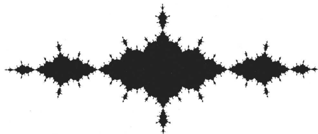
waarbij z de complexe notatie is van het punt met coördinaten (x,y) en c correspondeert met (a,b). In de gewone schrijfwijze van een dynamisch systeem is dat hetzelfde als

(1.3)
$x' = x^2 - y^2 + a$
$y' = 2xy + b$

In de, nogal wiskundige, theorie van de Julia-verzamelingen is de compacte notatie (1.2) het handigst, maar wanneer we zo'n set met de computer willen berekenen, moeten we de uitvoeriger formulering van (1.3) gebruiken. In dit geval is door (1.2) voor elke waarde van c, dat wil zeggen voor elk punt van het platte vlak, een dynamisch systeem gedefinieerd en behoort hierbij een Julia set. Die set blijkt met twee uitzonderingen een echte fractal te zijn. De eerste uitzondering is $c = 0$ (dat wil zeggen $a = b = 0$), waarvoor de Julia set een cirkel is. De tweede uitzondering is $c = -2$ (of $a = -2$, $b = 0$), in welk geval de Julia set een lijnstuk blijkt te zijn. Door (1.2), of liever door (1.3), wordt een dynamisch systeem bepaald waarbij de banen weer naar een of andere aantrekker toegaan. Helaas levert dat weinig op: of de banen verdwijnen in het oneindige, of ze gaan naar een of meer punten toe. Het meest interessante object is echter de zogenaamde 'afstoter', de verzameling waar alle naburige banen van wegvluchten. Het is de door Julia beschreven figuur, de Julia-fractal. Om een Julia set, of althans een goede benadering, op het beeldscherm van een computer te krijgen, is veel vernuft nodig. Maar hier brengt de Monte Carlo-methode redding. We vervangen het zojuist beschouwde dynamische proces door het inverse proces, dat wil zeggen we verwisselen eenvoudig opvolger en voorganger. In formulevorm volgt uit (1.2)

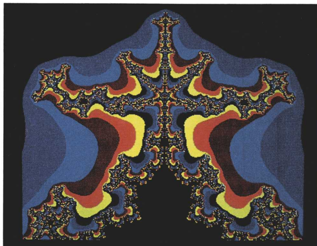
(1.4)
$z = +\sqrt{z' - c}$ of $z = -\sqrt{z' - c}$

Het omgekeerde proces is dus tweewaardig, dat wil zeggen we hebben bij elke stap twee mogelijkheden. We verkeren nu in een soortgelijke situatie als bij de Von Koch-lijn die door twee contracties L en R bepaald is. In plaats van de twee gelijkvormigheidstransformaties L en R hebben we nu twee ingewikkelder transformaties (1.4), die meetkundig moeilijker te doorzien zijn. We komen er later in de hoofdstukken 11 en 12 over de Julia sets nog op terug. Hier volstaan we met de afbeelding van een enkele, zogenaamde 'gevulde' Julia-fractal volgens (1.2) met $c = -1.2$ (zie figuur 1.10). De illustratie is verkregen met behulp van het programma JULFILL dat zich achterin het boek bevindt.



Figuur 1.10 Een Julia-fractal

Ook hieraan is een heel hoofdstuk gewijd. Het valt wellicht op dat we het meervoud gebruiken, en inderdaad zijn er meer Mandelbrot sets, heel veel zelfs. In beginsel behoort bij elke familie van Julia-fractals een speciale Mandelbrot set. Onder een familie Julia sets verstaan we de sets die bepaald zijn door een formule als (1.2) of (1.3), waarin nog een onbepaalde complexe constante c of twee reële getallen a en b voorkomen. Anders gezegd, een zodanige familie dat bij elk punt (a,b) van het vlak een lid van de familie behoort. De Mandelbrot set is eigenlijk een soort staalkaart die ons vertelt welke typen familieleden er zijn. Sommige Julia sets bestaan uit losse punten, hangen als los zand aan elkaar en worden wel als stoffractals beschreven. Andere Julia sets vormen juist een sterk samenhangend geheel. De meest bekende Mandelbrot set behoort bij de kwadratische functie (1.2) en ziet eruit als een 'appelmannetje', zoals kinderen uit een grote appel en een paar kleintjes een soort poppetje weten te maken. De rand van de Mandelbrot set blijkt een buitengewoon gecompliceerde structuur te hebben, en blijkt bij herhaalde vergroting welhaast steeds ingewikkelder te worden. De Mandelbrot set is dan ook allesbehalve een zelfgelijkvormige fractal. Details van de rand kunnen met de pixelmethode op fraaie wijze zichtbaar worden gemaakt. De enige beperkingen zijn eigenlijk rekensnelheid en numerieke nauwkeurigheid. Natuurlijk kunnen met een supercomputer superplaatjes gemaakt worden, maar met eenvoudiger middelen valt toch heel wat te bereiken. Om op een beeldscherm van 300 000 pixels een biljoenvoudige vergroting (dus miljoen maal miljoen) van een detail van de rand van de Mandelbrot set te kunnen verkrijgen, is met een moderne computer thuis best te verwezenlijken. Figuur 13.4 toont een dergelijk detail.



Een detail van de Mandelbrot set van $z^4 + c$

De fractaldimensie

In het vorige hoofdstuk hebben we al gezegd dat de fractals hun naam te danken hebben aan de omstandigheid dat ze, meestal, een gebroken dimensie bezitten. Immers, 'fractal' is het Engelse equivalent van ons woord 'gebroken'. In de oorspronkelijke opvatting van Mandelbrot was de eigenschap van het hebben van een gebroken dimensie allesoverheersend, maar nu denken we daar iets anders over. In de eerste plaats zijn er fraaie figuren die eruit zien als fractals, maar die toch een gehele dimensie zoals 1 of 2 hebben, en die zouden we niet graag willen uitsluiten. Verder is het in veel gevallen moeilijk, zo niet onmogelijk, om voor een gegeven fractal de fractaldimensie echt te bepalen. Zo weten we nog steeds niet hoe groot de dimensie van de rand van de Mandelbrot set is. Het is kennelijk een heel lastig wiskundig probleem, maar er zijn wel sterke aanwijzingen dat die dimensie 2 is. Voor wie sterk vergrote details van die rand op zijn beeldscherm weet te krijgen is dat inderdaad een heel plausibel vermoeden. Het lijkt wel of bij elke schaalvergroting nog grilliger patronen kunnen ontstaan, welke de indruk wekken dat het randgebied steeds dichter met lijntjes wordt bedekt. In het hoofdstuk over de Mandelbrot set en elders in dit boek kunnen we daar werkelijk indrukwekkende voorbeelden van zien. Het zou jammer zijn wanneer we zouden moeten zeggen dat de rand van de Mandelbrot set geen fractal zou zijn. Kennelijk zag Mandelbrot de bui al hangen, want later gaf hij een ruimere definitie van het begrip fractal, zij het dat de nieuwe definitie zo ruim was, dat het in streng wiskundige zin nauwelijks de naam van definitie verdiende.
Om een aantal redenen is het belangrijk om aan een gecompliceerde meetkundige figuur een fractaldimensie te kunnen toekennen. In de wereld om ons heen, bij praktische problemen van natuur en techniek, komen talloze grillige patronen voor die men gaarne zou willen karakteriseren of classificeren met een of meer getallen. We kunnen daarbij denken aan een wolkenlucht, schapenwolkjes bijvoorbeeld, een patroon met een mengeling van regelmaat en chaos. De weerman zegt op grond van zijn ervaring dat het misschien gaat regenen, of dat het toch mooi weer blijft. Een meteoroloog zou wellicht daaraan een fractaldimensie willen verbinden om op grond van het numerieke resultaat tot een conclusie te kunnen komen. Het zou tot een volledig geautomatiseerde weersvoorspelling kunnen komen, waarbij de computer al het werk van begin tot einde doet.

Dimensie, een veelzijdig begrip

In de wiskundeliteratuur komt het begrip dimensie in verschillende varianten voor. De zogenaamde topologische dimensie is de eenvoudigste. Die is gewoon 0 voor een punt, 1 voor een lijn en 2 voor een vlak. Beschouwen we in het platte vlak bijvoorbeeld een cirkelschijf, dan heeft die de dimensie 2, de cirkelrand heeft de dimensie 1 en het middelpunt de dimensie 0. Daarnaast bestaan er vele andere definities waarbij allerlei tussengelegen waarden aangenomen kunnen worden. In de wiskundige literatuur wordt de meeste aandacht besteed aan de zogenaamde Hausdorff-dimensie, door de wiskundige Hausdorff in 1919 beschreven. Aan de definitie zit wiskundig te veel vast om deze hier te kunnen bespreken. We beperken ons daarom tot een eenvoudiger versie, welke door Kolmogorov in 1958 geïntroduceerd werd, de zogenaamde *'box'dimensie* of *capaciteit*. We stellen ons voor dat zich in een vierkant met zijde 1, waarbij een willekeurige eenheid van lengte gekozen is, een fractal bevindt waarvan we de dimensie willen bepalen. We verdelen het vierkant door halvering in vier kleinere vierkanten en tellen het aantal vierkanten waarin zich punten van de fractal bevinden. Daarna halveren we de kleinere vierkanten, totdat er zestien vierkantjes zijn ontstaan met elk een zijde van 1/4. We bepalen weer het aantal waarin zich punten van de fractal bevinden. Zo gaan we verder met steeds kleinere 'box'jes. Bij de boekhouding vermelden we telkens de grootte van de zijde van het kleinste vierkantje en het aantal N die punten met de fractal gemeen hebben. Laten we aannemen dat het volgende fictieve lijstje is ontstaan, althans voorzover het de eerste drie kolommen betreft. De betekenis van de laatste kolom zal hieronder blijken.

rangnummer k	lengte zijde	aantal $N(k)$	D
0	1	1	
1	1/2	3	1.585
2	1/4	8	1.500
3	1/8	23	1.508
4	1/16	65	1.506
5	1/32	191	1.515
6	1/64	570	1.526
7	1/128	1692	1.523

We zien op het oog dat bij elke stap het aantal $N(k)$ ruwweg met de factor 3 toeneemt, terwijl de lengte telkens met de factor 2 verkleind wordt. De fractaldimensie in de zin van de 'box'dimensie kan dan eenvoudig benaderd worden als de limietwaarde van de logaritmen van $N(k)$ en van de reciproke zijden, dat wil zeggen

$$\text{fractaldimensie} \approx \log(\text{aantal})/\log(1/\text{zijde})$$

Traditiegetrouw wordt voor de fractaldimensie vaak de letter D gebruikt. In iets meer wiskundige vorm kan dezelfde formule dan geschreven worden als

(2.1)
$D \approx \log(N(k)) / (k \log(2))$

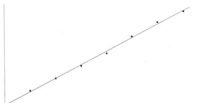
Wie van opschieten houdt en het oorspronkelijke vierkant liever in bijvoorbeeld honderd kleinere vierkantjes wil verdelen, en vervolgens de kleinste zijde telkens met een factor 10 verkleint, moet de bovenstaande formule een beetje aanpassen en vervangen door

$$D \approx \log(N(k)) / (k \log(10))$$

De waarden van het rechterlid van (2.1) zijn vanaf $k = 2$ in de vierde kolom van bovenstaand lijstje vermeld. We lezen hieruit onmiddellijk af dat de fractaldimensie ongeveer gelijk is aan 1.52. Een meer precieze waarde is in praktische gevallen nauwelijks mogelijk. Alleen wanneer het gaat om een door een wiskundige formule bepaalde fractal kan de fractaldimensie exact bepaald worden.

Een grafische bepaling

Sommigen werken liever met grafieken. Wie dat prettig vindt en ook vertrouwd is met logaritmisch papier doet er verstandig aan $N(k)$ logaritmisch uit te zetten tegen het rangnummer k. Elk punt van de vierde kolom correspondeert dan met een punt als in figuur 2.1. Op de horizontale as is k lineair uitgezet, op de verticale as is $N(k)$ uitgezet op logaritmische schaal. De verkregen puntenrij ligt dan ongeveer op een rechte lijn. De helling van die lijn is dan gelijk aan de fractaldimensie, mits de eenheden van de horizontale en de verticale schaalverdeling goed op elkaar zijn afgestemd. Uit (2.1) volgt namelijk



Figuur 2.1 De grafische bepaling van de fractaldimensie

$$\log(N(k)) = D(k \log(2))$$

de bekende vergelijking van een rechte lijn door de oorsprong wanneer $k \log(2)$ met x en $\log(N(k))$ met y correspondeert.
Het is altijd verstandig om bij het werken met een nieuwe definitie na te gaan of ook de oude vertrouwde resultaten geldig blijven. We verwachten dat een gevuld vierkant, in andere woorden de verzameling van punten die binnen een vierkant liggen, natuurlijk de dimensie 2 heeft. Wanneer we formule (2.1) toepassen, dan is voor elk rangnummer k het aantal overdekkende vierkantjes gelijk aan 4^k, tegelijk het totale aantal. Immers, alle vierkantjes bevatten wel punten van de gegeven verzameling. Formule (2.1) leidt dan meteen tot het antwoord $\log(4)/\log(2)$ ofwel 2, zoals het ook behoort.
Tot dusverre is de fractaldimensie bepaald met een soort ruitjespapier. Men zou zich kunnen voorstellen dat de fractal overdekt wordt door een transparant vel ruitjespapier waarvan de ruitjes steeds verder onderver-

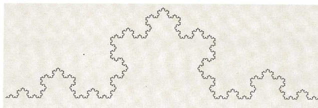
deeld kunnen worden. Maar theoretisch zou die werkwijze verbeterd kunnen worden door de ruitjes als het ware los te knippen, door de fractal zo goed mogelijk te overdekken door een aantal vierkantjes die natuurlijk allemaal wel dezelfde grootte moeten hebben. Die vierkantjes kunnen we dan naar believen verschuiven en verdraaien, net zo lang totdat we de fractal met zo weinig mogelijk overdekt hebben. Laten we de zijde van de 'box'jes h noemen en het optimale aantal vierkantjes $N(h)$. We kunnen in dat geval (2.1) vervangen door

(2.2)
$D \approx \log(N(h))/\log(1/h)$

een formule die des te beter geldt naarmate h kleiner is.

De fractaldimensie van de Von Koch-lijn

Het is een aardige opgave om met behulp van die formule de fractaldimensie van de Von Koch-lijn, afgebeeld in figuur 2.2, eens uit te rekenen. We bedenken dat de constructie van die lijn erop berust dat we telkens het middelste derde gedeelte vervangen door de opstaande zijden van een gelijkzijdige driehoek. Van een recht lijnstuk wordt telkens een gebroken lijn van vier kleinere lijnstukjes gemaakt. Na oneindig veel van die constructiestappen bestaat de Von Koch-lijn uit vier identieke stukken, waarvan elk een kopie is van het origineel met een verkleiningsfactor van 1/3. Laten we veronderstellen dat we werken met kleine vierkantjes, waarvan de zijde h is en dat er daarvan N nodig zijn. Verkleinen we h met de factor 3, dan kunnen we blijkbaar precies een van de vier kleinere kopieën van de Von Koch-lijn overdekken. Om alles te overdekken, hebben we $4N$ vierkantjes nodig. Verkleining van h met de factor 3 geeft dus een vergroting van N met de factor 4. Wie een beetje ervaring heeft opgedaan, trekt hieruit meteen de conclusie dat de fractaldimensie van de Von Koch-lijn gelijk is aan log(4)/log(3), dus aan 1.26. Wie er graag nog een beetje aan wil rekenen, zou kunnen opmerken dat h afneemt volgens een meetkundige reeks met reden 1/3. We kunnen dan h gelijk stellen aan $h = h_0 3^{-k}$, waarbij k het aantal verkleiningen is na het begin. Evenzo neemt N toe volgens een meetkundige reeks met reden 4, en kunnen we N gelijkstellen aan $N_0 4^k$. Wanneer we die uitdrukkingen substitueren in formule (2.2) en wanneer we aan k steeds grotere waarden toekennen, blijkt D inderdaad tot de vermelde waarde te naderen.



Figuur 2.2 De lijn van Von Koch

Een algemene formule

Voor zelfgelijkvormige fractals kan de fractaldimensie met behulp van een eenvoudige formule snel gevonden worden. In volgende hoofdstukken hebben we vaak met zulke fractals te maken. Het principe is als volgt. Voor dergelijke fractals bestaat er een aantal, zeg m, gelijkvormigheidstransformaties, zogenaamde contracties, welke de fractal transformeren in een deel van zichzelf. Een contractie is eenvoudig een transformatie waarbij alle afstanden verkort worden. Worden bijvoorbeeld de punten P en Q getransformeerd in P' en Q', dan is de lengte van het lijnstuk $P'Q'$ altijd kleiner dan die van PQ. De verhouding $P'Q'/PQ$ kunnen we de plaatselijke contractiefactor noemen. Het kan zijn dat die factor overal hetzelfde is, maar in het algemeen hoeft dat niet. De bovengrens f van alle plaatselijke contractiefactoren heet de *contractiefactor* van de gelijkvormigheidstransformatie. Voor willekeurige punten geldt dan altijd

$$P'Q' \leq fPQ$$

De contractiefactoren van de m gelijkvormigheidstransformaties waarvoor de fractal zelfgelijkvormig is, noemen we $f_1, f_2, f_3, \ldots, f_m$. De Hausdorff-dimensie D van de fractal is dan bepaald door de vergelijking

(2.3)
$$f_1^D + f_2^D + \ldots + f_m^D = 1$$

Overigens moet die formule wel met een beetje verstand gebruikt worden. De maximale dimensie van een vlakke figuur is 2, zodat een eventueel door (2.3) bepaalde hogere waarde meteen afgetopt moet worden en vervangen door 2.
Ter illustratie beschouwen we nog eens de Von Koch-lijn. We zien aan de figuur dat de hele lijn bestaat uit vier kleinere kopieën van zichzelf. De verkortingsverhouding is 1/3. Dus

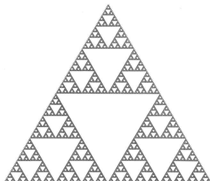
$$(1/3)^D + (1/3)^D + (1/3)^D + (1/3)^D = 1$$

Een eenvoudige berekening geeft snel het goede antwoord

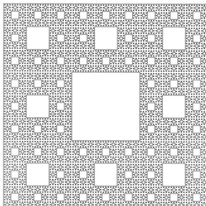
$$D = \log(4)/\log(3) = 1.26$$

Zeven van Sierpinski en van Menger

Vervolgens toetsen we de formule op de bekende zeef van Sierpinski geïllustreerd in figuur 2.3.
Er zijn verschillende manieren om deze figuur op het beeldscherm te krijgen. Het programma SIERLINE berust op een talstelselmethode waarbij volgens de structuur van het drietallig talstelsel steeds kleinere driehoeken aan de reeds gevormde figuur worden toegevoegd. Het programma SIERMC laat de zeef ontstaan als een stochastische wolk van punten gevormd volgens de zogenaamde Monte Carlo-methode, die in hoofdstuk 6



Figuur 2.3 De zeef van Sierpinski



Figuur 2.4 De spons van Menger

wordt besproken. Wanneer we de figuur ten opzichte van een van de drie hoekpunten verkleinen met de factor 1/2, een centrale vermenigvuldiging, dan gaat de Sierpinski-fractal over in een deel van zichzelf. Er zijn aldus drie contracties, elk met de factor 1/2. De formule geeft dan

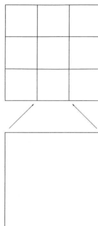
$$(1/2)^D + (1/2)^D + (1/2)^D = 1$$

zodat

$$D = \log(3)/\log(2) = 1.58$$

In figuur 2.4 tonen we de zogenaamde zeef of spons van Menger, een vierkant analogon van de zeef van Sierpinski. In het beroemde boek van Mandelbrot kan men een illustratie vinden van de 'spons'versie, een driedimensionale constructie waarbij uit een kubus telkens kleinere kubusjes worden weggenomen. We beperken ons hier echter tot het tweedimensionale equivalent, de 'zeef'versie, die men trouwens met weinig moeite op het computerscherm kan krijgen. Men kan daartoe het programma MENGERMC, waarbij de Monte Carlo-methode gebruikt is, of MENGERCL, het laatste in kleur, gebruiken. We tellen in totaal acht contracties, centrale vermenigvuldigingen, elk met de factor 1/3, waarvoor de fractal zelfgelijkvormig is (figuur 2.5).
Formule (2.2) geeft dan op analoge wijze meteen het resultaat

$$D = \log(8)/\log(3) = 1.89$$



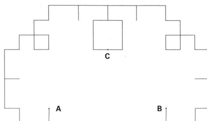
Figuur 2.5 De acht contracties van de spons van Menger

De lijn van Lévy als grensgeval

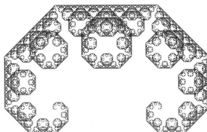
Het is aardig eens een grensgeval te onderzoeken, waarbij de formule precies $D = 2$ geeft. De eenvoudigste situatie ontstaat voor twee contracties, elk met de factor $1/\sqrt{2}$. Dat is het geval bij de bekende lijn van Lévy, ook wel *C*-kromme genoemd. Een lage benadering van de fractal, verkregen met behulp van het programma LEVYLINE, is afgebeeld in figuur 2.6.
Zoals we zien is het linker eindpunt *A* het centrum van een gelijkvormigheidstransformatie, waarbij het rechter eindpunt *B* overgaat in het punt *C*. Die transformatie is de combinatie van een draaiing over 45 graden en een verkorting met de factor $AC/AB = 1/\sqrt{2}$. Evenzo is het rechter eindpunt het centrum van een overeenkomstige contractie, waarbij het lijnstuk *BA* overgaat in *BC*. De dimensievergelijking is hier

$$(1/\sqrt{2})^D + (1/\sqrt{2})^D = 1$$

Inderdaad is voor deze fractal $D = 2$, een resultaat dat we kunnen interpreteren door te zeggen dat de lijn van Lévy lokaal vlakvullend is. In figuur 2.7 tonen we een veel hogere benadering van deze fractal, en inderdaad blijken de punten van de fractal zich plaatselijk steeds meer op te hopen. Bij een dergelijke hoge benadering komen de gaten sterk naar voren. Uiteraard zijn al die gaten onderling gelijkvormig ten opzichte van de twee basistransformaties. Aan figuur 2.7 ligt het programma LEVYLONG ten grondslag, een versie van LEVYLINE waarbij zogenaamde 'long integers' kunnen worden gebruikt.
Op overeenkomstige wijze heeft ook de in figuur 2.8 gegeven drakenkromme de fractaldimensie 2. Het desbetreffende programma is DRAGON. Men kan langs wiskundige weg vrij eenvoudig laten zien dat de ideale fractal bestaat uit een serie van onderling gelijkvormige, vlakke, massieve 'tegels' van een grillige vorm.



Figuur 2.6 Een lage benadering van de lijn van Lévy



Figuur 2.7 Een hoge benadering van de lijn van Lévy



Figuur 2.8 De drakenlijn

Multifractals

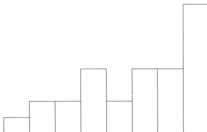
Bepaalde toepassingen als de verdeling van materie hebben Mandelbrot aanleiding gegeven het begrip *multifractal* in te voeren. Goud wordt op aarde slechts op enkele plaatsen in een hoge concentratie aangetroffen, maar op meer plaatsen kan het in een lagere concentratie gevonden worden. Letten we op goudklompjes met een bepaalde concentratie aan goud, dan kan men zich voorstellen dat ze verdeeld zijn volgens een fractalpatroon en dat daarbij dan een fractaldimensie behoort. Nemen we materie met een hogere concentratie aan goud, dan zouden we een andere fractal vinden met ook een andere concentratie. De totaliteit van het goud, wellicht ook weer aanwezig in een fractalpatroon, is dus als het ware opgedeeld in een familie van fractals, elk met een eigen dimensie. We spreken dan van een multifractal, een begrip dat niet alleen voor de geologie, maar ook bij andere toepassingen nuttig is gebleken.
Een multifractal op een rechte lijn kan men dus opvatten als een massaverdeling van de lijn. Laten we ons voor het gemak even beperken tot een massief lijnstuk dat zich in coördinaten uitstrekt van 0 tot 1. We beschouwen onszelf als een zilversmid die een staaf zilver door drijven moet bewerken. In de edelsmeedkunst is drijfwerk een techniek om door massaverplaatsing het metaal in een bepaalde vorm te kunnen brengen. In dit uiterst eenvoudige voorbeeld blijft de staaf weliswaar een staaf van dezelfde lengte, maar door hameren brengen we zoveel metaal van de linkerhelft naar de rechterhelft over, dat de linkerhelft de massa p en de rechterhelft de massa $1-p$ heeft. Elke helft is homogeen van samenstelling. Wie graag een concrete situatie wil, neme $p=1/3$, zodat de massa's links en rechts in de verhouding $1:2$ verdeeld zijn.

De uitwerking in het tweetallig talstelsel

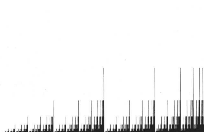
Men kan er een eenvoudig grafiekje van maken zoals in figuur 2.9. We herhalen het beschreven proces op recursieve wijze door het op elk van de twee stukken toe te passen, enzovoort. Het volgende schema laat zien hoeveel massa telkens elk gedeelte van de staaf heeft. Ter wille van de eenvoud schrijven we q in plaats van $1-p$.

stap 1	p	q						
stap 2	p^2	pq	pq	q^2				
stap 3	p^3	p^2q	p^2q	pq^2	p^2q	pq^2	pq^2	q^3
stap 4	p^4	p^3q	p^3q	p^2q^2	...			

In het volgende plaatje (figuur 2.10) tonen we de massaverdeling na twaalf stappen.
Het valt op hoe grillig de massa nu reeds verdeeld is, en dan te bedenken dat dit maar de eerste paar stapjes zijn op weg naar de oneindigheid. Beide figuren zijn verkregen met het programma MULTIFR. Dat programma berust op de schrijfwijze van gehele getallen in het tweetallig talstelsel. Wanneer we de getallen van 0 tot 2^3 tweetallig schrijven, krijgen we het rijtje



Figuur 2.9 Fractale massaverdeling in een staaf na drie stappen



Figuur 2.10 De massaverdeling na twaalf stappen

000 001 010 011 100 101 110 111

Vergelijken we dit rijtje met het overeenkomstige rijtje bij stap 3, dan zien we een opvallende overeenstemming. Het aantal nullen correspondeert met een zelfde aantal factoren p, terwijl het aantal enen de factoren q geeft. Het is dus niet moeilijk te voorspellen hoeveel massa er op bijvoorbeeld de achtste stap terechtkomt op een willekeurige plaats in de staaf. In totaal zijn er $2^8 = 256$ stukjes. Voor het element met rangnummer 89 hebben we in tweetallige notatie $89 = 64 + 16 + 8 + 2 + 1$, dus

$89 = 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1$

De hoeveelheid massa is derhalve p^3q^5. Het programma is dus niets anders dan voor elk geheel getal tot aan een bepaalde grens, zeg 2^{12}, de binaire ontwikkeling te bepalen en op grond van het aantal nullen in die ontwikkeling een lijntje te tekenen.

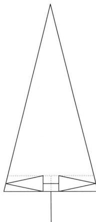
Op overeenkomstige wijze kunnen multifractals in vlak en ruimte gedefinieerd worden. De toepassingsmogelijkheden nemen daarbij ook toe, maar het onderzoek daarover staat nog in de kinderschoenen. Ook al omdat het onderwerp nogal technisch van aard is kunnen we er niet veel meer over zeggen. Men kan onder andere denken aan landschappen en aan wolkenluchten.

Zelfgelijkvormigheid

In de vorige twee hoofdstukken hebben we al verschillende malen te maken gehad met de zelfgelijkvormigheid van fractals, maar in dit hoofdstuk hebben we de gelegenheid er op meer systematische wijze over te praten. Voorop staat dat de zelfgelijkvormigheid het essentiële kenmerk is van een fractal. Die zelfgelijkvormigheid kan zich manifesteren op een regelmatige en streng wiskundige wijze, zoals bij de lijn van Von Koch, de zeef van Sierpinski of de Pythagoras-boom. Het kan echter ook zijn dat de zelfgelijkvormigheid meer in toevallige zin, volgens de wetten van de waarschijnlijkheidsrekening, opgevat moet worden. We kunnen daarbij denken aan een wolkenlucht of aan een echte boom.
In het boek *Fractals* hebben we laten zien hoe de wiskundige zelfgelijkvormigheid vastgelegd kan worden in meetkundige transformaties. In het bijzonder hebben we in dat boek een aantal fractals daadwerkelijk geconstrueerd door uit te gaan van twee gegeven gelijkvormigheidstransformaties. In dit boek leggen we het accent echter op het omgekeerde probleem. Stel dat we een mooie, fractalachtige figuur zien, of dat we in de natuur een patroon zien dat ons doet denken aan een fractal. De vraag rijst dan of we een soort wiskundig mechanisme kunnen bedenken dat ons in staat stelt een dergelijke figuur als een zelfgelijkvormige fractal te synthetiseren. In een aantal gevallen is dat niet eens zo moeilijk, wanneer men tenminste geen nauwkeurige gelijkenis nastreeft. Het hangt er maar vanaf hoeveel inzicht men heeft in meetkundige transformaties. In dit hoofdstuk zullen we trachten de lezer daartoe de nodige meetkundige kennis te verschaffen zonder daarbij op wiskundige details in te gaan. Uiteraard ontkomen we niet aan een beetje wiskundige techniek, maar wie nog wat schoolse kennis bezit over zaken als gelijkvormige driehoeken, en wie als menige computergebruiker kan werken met x,y-coördinaten in een plat vlak, zou het desnoods met een beetje moeite wel kunnen begrijpen.

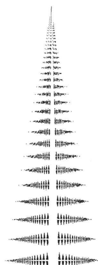
Het varenblad van Barnsley

Om de gestelde problematiek duidelijk te maken, beschouwen we een concreet voorbeeld, ons aangereikt door de wiskundige Barnsley die zijn boek *Fractals Everywhere* (een mooi boek met veel 'moeilijke' wiskunde) opgesierd heeft met een paar synthetische varenbladeren.
In figuur 3.1 zien we de hoofdvorm van een varenblad, een smalle, gelijkbenige driehoek ABT met een steeltje. Aan de symmetrieas is een groot aantal zijbladeren gehecht die langzaam van beneden naar boven in groot-



Figuur 3.1 Het schema van een varenblad

te afnemen. In de figuur zijn daarvan alleen de onderste twee in symmetrische positie geschetst. Elk zijblad is dus een verkleinde kopie van de hoofdvorm. Het grootste zijblad linksonder, $A_1B_1T_1$, is een driehoek die gelijkvormig is met de grote driehoek ABT. Men kan zich voorstellen dat de kleine driehoek uit de grote ontstaan is door verkleining en draaiing. De mate van verkleining is bepaald door de verhouding A_1B_1/AB. Dat getal noemen we de reductiefactor of de contractiefactor. In de figuur is een factor 1/5 aangenomen. Verder zien we dat de kleine driehoek ten opzichte van de grote over een hoek van 90 graden naar links, positief gerekend tegen de wijzers van een klok in, gedraaid is.
Op overeenkomstige wijze is er een tweede gelijkvormigheidstransformatie die uit driehoek ABT het grootste zijblad rechtsonder afleidt. De kenmerken zijn dezelfde reductiefactor en een rotatiehoek van −90 graden, negatief gerekend in wijzerzin. Ten slotte kunnen we nog een derde gelijkvormigheidstransformatie onderkennen, en wel een centrale vermenigvuldiging met centrum T. Wanneer we de figuur ten opzichte van de top T een klein beetje verkleinen, met de factor 11/12 bijvoorbeeld, dan gaat het varenblad ABT over in een iets kleiner deel van zichzelf als geschetst in de figuur.
Natuurlijk zijn er nog veel meer gelijkvormigheden aanwezig. Elk zijblad draagt weer een reeks van kleinere zijbladen, en die bevatten op hun beurt weer..., precies zoals we dat in de natuur kunnen waarnemen. Men kan echter, eventueel gesteund door wiskundige ervaring, constateren dat die gelijkvormigheden slechts combinaties van de genoemde drie zijn. Op het computerprogramma FERN1 gaan we nu niet in, maar we volstaan met de mededeling dat de wiskundige kern berust op herhaald gebruik van de genoemde drie transformaties. Het resultaat is figuur 3.2 die al aardig op een echt varenblad begint te lijken, maar waaraan nog de steeltjes ontbreken.

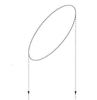


Figuur 3.2 Een synthetisch varenblad

Projecteren

Wanneer we aan de in figuur 3.2 afgebeelde fractal nog steeltjes willen toevoegen zodanig dat de uitgebreidere fractal nog beter op een echt varenblad gaat lijken, dan moet er een vierde transformatie aan toegevoegd worden, een meetkundige vervorming die 'steeltjes' maakt van 'bladeren'. Wie een beetje met meetkundige zaken vertrouwd is, begrijpt onmiddellijk dat de gewenste transformatie een soort projectie moet zijn. In het algemeen is een projectie een meetkundige bewerking die een ruimtelijke figuur reduceert tot iets vlaks, die een vlakke figuur terugbrengt tot een lijn, die een lijn transformeert in een punt. Projecteren is een transformatie waardoor de (topologische) dimensie van een figuur verlaagd wordt. Het is zoiets als schaduw werpen. In figuur 3.3 laten we zien hoe een bladvormige figuur door projectie wordt getransformeerd in een lijnstukje, precies wat we nodig hebben bij het varenblad.
De eenvoudigste vlakke projectievorm is die waarbij alle punten in het vlak terechtkomen op een vaste rechte lijn. De meest gebruikelijke wijze is die waarbij punten loodrecht op die lijn worden geprojecteerd. Het is duidelijk dat projecteren een onomkeerbaar proces is, uit de projectie

kunnen we nooit het origineel terugvinden. De projectie is als meetkundige transformatie dus niet omkeerbaar. In de coördinatenmeetkunde–we komen er aanstonds op terug–kan projecteren opgevat worden als het weglaten van een coördinaat. Het punt (x,y) in het vlak wordt bijvoorbeeld vervangen door de projectie $(x,0)$ op de x-as.
Projectie kan opgevat worden als een contractie, als een afbeelding waarbij gecontraheerd of verkleind wordt. Alleen is de contractiefactor gewoon nul. In figuur 3.3 zien we dan ook dat een blad met een zekere oppervlakte 'platgeslagen' wordt tot een lijnstuk dat geen oppervlaktemaat bezit, of waarvan, anders gezegd, het oppervlak nul is. Men kan zich voorstellen dat het niet moeilijk is aan het computerprogramma met de drie gelijkvormigheidstransformaties nog als vierde een projectie toe te voegen waarbij het blad van figuur 3.1 in horizontale richting op de verticale hoofdsteel is geprojecteerd. Het resultaat is figuur 3.4, een heel wat betere imitatie. Het desbetreffende programmatype FERN2 zal in hoofdstuk 6 besproken worden.



Figuur 3.3 De projectie van een vlakke figuur

Transformaties

Om inzicht te krijgen in de structuur van fractals is enige kennis van meetkundige transformaties onontbeerlijk. Ten gerieve van de lezer zetten we hier een aantal nuttige zaken op een rijtje, waarbij we trachten met zo weinig mogelijk wiskundige techniek uit te komen. Men verwachte dus geen afleidingen van formules. Voor wie zelf fractalprogramma's wil samenstellen of wie de hier geboden programma's geheel wenst te begrijpen, is enig gebruik gemaakt van coördinatenmeetkunde.
Een transformatie is een vervorming van het platte vlak waarbij punten verplaatsingen ondergaan. We spreken af dat P een nog niet verplaatst punt aangeeft, het zogenaamde *origineel*, en dat P' het verplaatste punt is, het *beeldpunt* of beeld van P. Transformaties duiden we vaak aan met hoofdletters als T, U, V, L, R enzovoort. Dat de transformatie T het punt P in P' transformeert kunnen we symbolisch heel kort noteren als $P' = \mathrm{T}(P)$. Het is handig een aparte notatie te hebben voor de transformatie die alles op zijn plaats laat. We schrijven er I voor, met gebruikmaking van de beginletter van het woord 'identiek'. Immers, beeld en origineel zijn identiek, en symbolisch geldt $\mathrm{I}(P) = P$ voor alle punten P. Men noemt I de identieke transformatie of kortweg de *identiteit*.
Transformaties kunnen na elkaar uitgevoerd worden, een bewerking die de kern van het fractalbegrip raakt. Zijn T en U gegeven transformaties, dan betekent de notatie $\mathrm{UT}(P)$ dat we P eerst onderwerpen aan de transformatie T en dat we het beeld van P weer transformeren met U. Het schema is dus

$$\mathrm{T}(P) = P'$$
$$\mathrm{U}(P') = P''$$

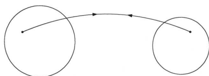
De gecombineerde transformatie UT (eerst T dan U) heet het *produkt* van T en U. Dat begrip heeft maar weinig te maken met het overeenkomstige vermenigvuldigen van getallen. Bij gewone getallen is de volgorde van de



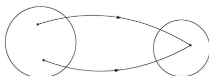
Figuur 3.4 Een beter varenblad

factoren niet van belang. Zo zijn 3×5 en 5×3 beide gelijk aan 15. Bij transformaties is de volgorde in de regel wel van belang, en kunnen TU (eerst U en dan T) en UT (T gevolgd door U) iets heel verschillends zijn. Gelukkig is dit het enige verschil met het vermenigvuldigen van getallen. Sommige transformaties kunnen omgekeerd worden. Dan behoort telkens één origineel bij een beeld. Uit het voorgaande blijkt al dat projecties dat zeker niet zijn. De hieronder te bespreken lineaire transformaties, welke in de praktijk het meeste voorkomen, zijn dat gelukkig wel, al kunnen ze bij wijze van uitzondering een projectie zijn. Is T omkeerbaar – de technische term is inverteerbaar – dan wordt de omgekeerde of inverse transformatie aangeduid als T^{-1}. De naam is kortweg de *inverse* van T. In de symbolische schrijfwijze hebben we $T^{-1}T = I$, een soort stenogram voor de mededeling dat wanneer een willekeurig punt P door T getransformeerd wordt naar P' en wanneer dat punt het origineel is van de inverse transformatie T^{-1}, het beeldpunt daarvan weer P is (zie figuren 3.5 en 3.6).
Wanneer een zelfde transformatie T almaar herhaald wordt, kunnen we de gebruikelijke notatie van machten gebruiken. Zo betekent T^3 gewoon dat de transformatie T driemaal herhaald wordt (figuur 3.7).
Zijn bijvoorbeeld L en R twee gegeven transformaties, dan betekent L^2RLR^3 een samengestelde transformatie waarbij een punt of een bepaalde figuur eerst driemaal onderworpen wordt aan de transformatie R. Het resultaat wordt daarna getransformeerd met behulp van L en dan met R. De nieuwe figuur wordt ten slotte tweemaal bewerkt met L. Het lijkt moeilijk een voorstelling te hebben van het uiteindelijke resultaat. Maar wanneer zowel L als R gelijkvormigheidstransformaties zijn, is elke uit de elementen L en R samengestelde combinatie dat ook.

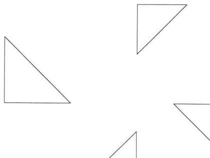
Bij een willekeurige transformatie T kan er een punt bestaan dat op zijn plaats blijft, dus een punt waarvoor $T(P) = P$. Een dergelijk punt noemt men een *dekpunt*. In de Engelse terminologie spreekt men van een *fixed point*, wat misschien wel zo sprekend is. Is T bijvoorbeeld een rotatie om een punt O als draaipunt, dan is O een dekpunt. Andere dekpunten zijn er niet. Is daarentegen T een spiegeling om een lijn a, de as van spiegeling, dan zijn alle punten van a dekpunten, omdat ze bij spiegeling op hun plaats blijven. Bovendien weten we dat een herhaalde spiegeling de oorspronkelijke toestand herstelt, symbolisch dus $T^2 = I$.
De voor de theorie van de fractals meest belangrijke transformaties zijn zogenaamde contracties. Bij een *contractie* T worden lijnstukjes altijd verkleind. Zijn P en Q willekeurige punten en P' en Q' hun beelden ten opzichte van T, dan is de verhouding $P'Q'/PQ$ een maat voor de verkorting. Soms hangt dat getal niet af van de positie van P en van Q en dan heet het per defi-



Figuur 3.5 Een omkeerbare transformatie



Figuur 3.6 Een niet-omkeerbare transformatie



Figuur 3.7 Herhaling van een transformatie

nitie de *contractiefactor* van T. Wanneer dat getal wel van de plaats van P en Q afhangt, kunnen we slechts spreken van een plaatselijke verkorting. De contractiefactor van T wordt dan gedefinieerd als de grootste waarde (voor wiskundigen de bovengrens) van alle plaatselijke verkortingen. Men lette erop dat de contractiefactor altijd betrekking heeft op de verkleining van lijnstukken en niet op die van oppervlakten. Heeft een contractie bijvoorbeeld overal dezelfde plaatselijke verkorting van 1/3, dan wordt het oppervlak van een vierkant of een cirkel tot 1/9 van de oorspronkelijke grootte gereduceerd.

Lineaire transformaties

Lineaire transformaties zijn gekenmerkt door de eigenschap dat rechte lijnen ook weer in rechte lijnen worden getransformeerd. Bovendien blijft ook de evenwijdigheid behouden, zodat evenwijdige lijnen ook na transformatie evenwijdig blijven. Een lineaire transformatie die werkt op een schaakbord of op ruitjespapier maakt er iets scheefs van, waarbij de vierkantjes vervormd zijn tot onderling even grote parallellogrammen als geschetst in figuur 3.8.
In x,y-coördinaten wordt een lineaire transformatie beschreven door

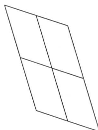
(3.1)
$x' = ax + by + e$
$y' = cx + dy + f$

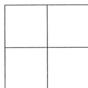
Dit zijn heel eenvoudige formules, die in veel programma's voorkomen. Bij lineaire transformaties wordt een oppervlak altijd met dezelfde factor vergroot of verkleind. Voor lijnstukjes daarentegen hangt de verandering van lengte af van de richting van het betrokken lijnstukje. We zien dat al gebeuren in figuur 3.8, waar bij een vierkantje de ene zijde verlengd en de andere zijde juist verkort wordt. De schaalfactor die ons zegt hoe een oppervlak door de transformatie (3.1) gewijzigd wordt, kunnen we bepalen met behulp van de uitdrukking

(3.2)
$ad - bc$

een uiterst nuttige formule. Komt er een negatieve waarde uit, dan heeft de transformatie het karakter van een spiegeling. De oppervlakteverandering moeten we dan zien als de absolute waarde van die uitdrukking. Is de uitdrukking nul, dan wordt een oppervlak letterlijk platgeslagen en is de transformatie een projectie.

Figuur 3.8 Een lineaire transformatie





In de praktijk van het ontwerpen van een fractal moeten de zes coëfficiënten van de transformatie (3.1) berekend worden op grond van meetkundige overwegingen. In zijn algemeenheid lijkt dat veel werk, het oplossen van zes onbekenden uit zes vergelijkingen, maar meestal valt het erg mee, zoals bij de talloze nog te bespreken voorbeelden zal blijken. In veel gevallen kunnen de formules (3.1) vervangen worden door eenvoudigere.
Is de transformatie (3.1) een draaivermenigvuldiging, dat wil zeggen een rechtstreekse gelijkvormigheidstransformatie die opgebouwd is uit een centrale vermenigvuldiging en een draaiing, dan kunnen we uitgaan van de formules

(3.3)
$x' = ax - by + e$
$y' = bx + ay + f$

Alle lijnstukken worden nu op dezelfde wijze verkleind of vergroot. De contractiefactor, of algemener de lineaire schaalfactor, kan berekend worden als

(3.4)
$s = \sqrt{a^2 + b^2})$

een resultaat dat onmiddellijk uit (3.2) volgt.
Is α de hoek waarover gedraaid wordt, dan geldt

(3.5)
$a = s\cos(\alpha)$
$b = s\sin(\alpha)$

met behulp waarvan de hoek α afgeleid kan worden. Zijn omgekeerd s en α bekend, dan kunnen de waarden van de coëfficiënten a en b meteen vastgesteld worden.
Is de transformatie (3.1) de combinatie van een centrale vermenigvuldiging en een spiegeling, dan zien de formules eruit als

(3.6)
$x' = ax + by + e$
$y' = bx - ay + f$

Ook hierbij worden alle lijnstukken vergroot of verkleind en geldt voor de schaalfactor dezelfde formule (3.4).
In de praktijk heeft men vaak te maken met speciale gevallen van (3.3) en (3.6). Voor een centrale vermenigvuldiging kan men de formules (3.3) gebruiken, maar omdat een draaiing daarbij afwezig is, kan men $b = 0$ nemen omdat de hoek α dan nul is. Dus

(3.7)
$x' = ax + e$
$y' = ay + f$

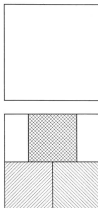
De factor a is daarbij dan positief en heeft de betekenis van contractiefactor. Dezelfde formules (3.7) kunnen we ook gebruiken wanneer de centrale vermenigvuldiging gepaard gaat met een puntspiegeling, of met een draaiing over 180 graden, wat hetzelfde is. In dat geval is de constante a negatief.
Een ander veel voorkomend geval is de kwartdraai waarbij α gelijk is aan 90 graden. Dan heeft men

(3.8)
$x' = -by + e$
$y' = bx + f$

waarbij de contractiefactor b positief is voor een kwartdraai in positieve zin, tegen de wijzers van de klok in, en negatief voor een kwartdraai met de wijzers mee.

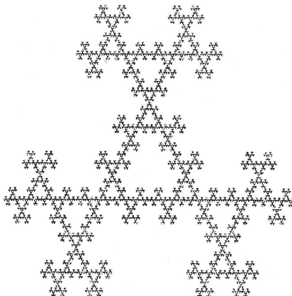
Voorbeeld

Wie zelf fractalprogramma's wil maken, kan met dit wiskundige materiaal werkelijk wonderen verrichten. In figuur 3.9 zien we een vierkant $ABCD$, waarin drie kleinere vierkanten geplaatst zijn. We stellen ons tot taak een zelfgelijkvormige fractal te ontwerpen die in zichzelf getransformeerd wordt door de gelijkvormigheidstransformaties die het grote vierkant in een der drie kleinere vierkanten overvoeren. Er zijn dus drie verschillende transformaties werkzaam, die we gemakshalve met L, R en T aanduiden. Elk van die transformaties verkleint het grote vierkant tot halve grootte, dat wil zeggen met de contractiefactor 1/2. Daarna hebben we nog de keus om het verkleinde vierkant al of niet gedraaid in de positie van een der kleinere vierkanten te brengen. Voor het programma moeten we coördinaten gebruiken, en die kiezen we natuurlijk zo eenvoudig mogelijk. De symmetrieassen van $ABCD$ nemen we als x- en y-as, en we geven de hoekpunten de coördinaten $A(-1,-1)$, $B(1,-1)$, $C(1,1)$, $D(-1,1)$.
In het programma SQUARE1 is een aantal verschillende mogelijkheden verenigd. De eenvoudigste keuze, waarbij L, R en T centrale vermenigvuldigingen zijn, blijkt wellicht tot verrassing van velen niets anders dan de zeef van Sierpinski te zijn, zij het met een kleine, verticale vervorming. Maar wanneer we het grote vierkant $ABCD$ gedraaid op de kleinere vierkanten afbeelden als aangegeven in het schema van figuur 3.9, dan krijgen we de in figuur 3.10 afgebeelde fractal, iets heel anders. Weer een andere keuze geeft figuur 3.11.
Wie nu de smaak te pakken heeft, kan nog andere combinaties kiezen. In elk van de drie gevallen zijn er inclusief spiegelingen acht manieren om te transformeren, totaal dus $8 \times 8 \times 8$ of 512 mogelijkheden. Dat zullen niet in alle gevallen even boeiende fractals zijn, en soms komt men bij een andere combinatie toch weer hetzelfde type tegen, maar er is ruimschoots gelegenheid om zelf te experimenteren. Het programma SQUARE2 biedt daartoe alle gelegenheid.

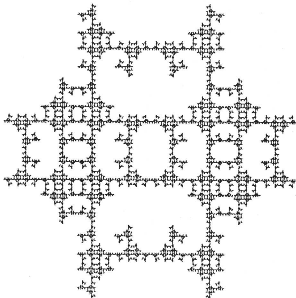


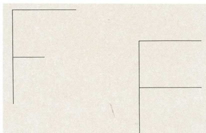
Figuur 3.9 Drie afbeeldingen van een vierkant

Figuur 3.10 De bijbehorende fractal

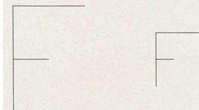


Figuur 3.11 Fractal van dezelfde familie

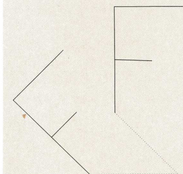




Figuur 3.12 Translatie



Figuur 3.13 Centrale vermenigvuldiging



Figuur 3.14 Rotatie

Overzicht

In deze alinea vatten we de verschillende lineaire transformaties nog eens schematisch samen, van eenvoudig tot ingewikkeld. In alle gevallen onderwerpen we de basisfiguur, een rechthoekige driehoek met zijden in de verhouding 2 : 1 : $\sqrt{3}$, een of meer keren aan dezelfde transformatie. Het dekpunt van de transformatie, voorzover van belang, geven we aan met een cirkeltje.

Translatie of parallelverschuiving (figuur 3.12)

Centrale vermenigvuldiging of schaling (figuur 3.13)

Rotatie of draaiing (figuur 3.14)

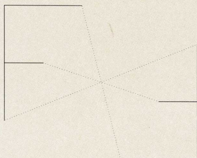
Puntspiegeling (figuur 3.15)
Zowel een speciaal geval van de centrale vermenigvuldiging als van de rotatie.

Draaivermenigvuldiging (figuur 3.16)
Combinatie van centrale vermenigvuldiging en rotatie.

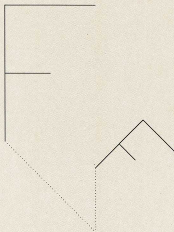
Lijnspiegeling (figuur 3.17)

Krimpspiegeling of *rekspiegeling* (figuur 3.18)
Combinatie van een lijnspiegeling en een centrale vermenigvuldiging. Bij fractals is alleen de krimpspiegeling, waarbij dus contractie plaatsvindt, van belang.

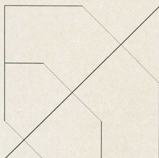
Lineaire vervorming (figuur 3.19)
De deftige naam is een affiene transformatie. Alle vorige transformaties waren gelijkvormigheden, maar bij de meer algemene affiene transformatie wordt ook de grootte van de hoeken veranderd.



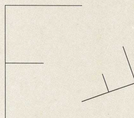
Figuur 3.15 Puntspiegeling



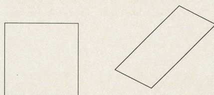
Figuur 3.16 Draaivermenigvuldiging



Figuur 3.17 Lijnspiegeling



Figuur 3.18 Krimp- of rekspiegeling

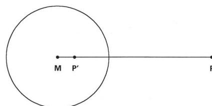


Figuur 3.19 Lineaire vervorming

Cirkelinversie

Er zijn nog veel meer interessante meetkundige transformaties waarmee fractals gemaakt kunnen worden. Men vat ze meestal samen onder de term *niet-lineaire* transformaties. Een belangrijk type zal aan de orde komen in het hoofdstuk over de Julia sets. Hier beperken we ons tot een zeer speciale niet-lineaire vervorming, de zogenaamde cirkelinversie. De cirkelinversie zou een leuk onderwerp voor de schoolmeetkunde kunnen zijn, omdat de theorie niet moeilijk is en de ermee bereikte resultaten werkelijk verbluffend zijn. Mandelbrot, de peetvader van de fractals, heeft er dan ook in zijn publikaties veel aandacht aan besteed.
Het principe kan zonder formules als volgt beschreven worden. Men denke zich in het vlak een cirkel *C* geplaatst. Alles wat zich daarbinnen bevindt noemen we de binnenwereld. Die binnenwereld denken we bevolkt met punten, rechte lijnstukjes en cirkelbogen. In de buitenwereld bevinden zich net zulke bewoners. Het doet denken aan het beroemde verhaal 'Platland' van Abbott, die een wereld van tweedimensionale wezens oproept die geen kennis hebben van een driedimensionale wereld. Hier hebben we echter te maken met twee werelden, een echte wereld en een schaduwwereld. De bewoners van de binnenwereld denken dat hun wereld de echte is, maar de bewoners van de buitenwereld hebben dezelfde opvatting. Ze hebben geen kennis van elkaars bestaan, maar wij hogere wezens, levend in drie dimensies, overzien beide werelden en weten dat beide werelden volledig equivalent zijn en dat ze met een eenvoudige meetkundige transformatie, de inversie, te verwisselen zijn. Bij die inversie blijven essentiële eigenschappen als de hoek tussen twee lijntjes bewaard.
De meetkundige inversie wordt aan de hand van figuur 3.20 gedefinieerd door de regel

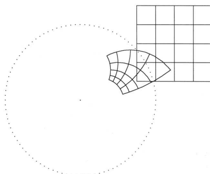
(3.9)
$MP\ MP' = R^2$



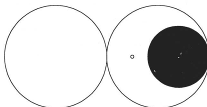
Figuur 3.20 De inversie van een punt ten opzichte van een cirkel

Ten opzichte van een gegeven cirkel met middelpunt M en straal R wordt een willekeurig punt P getransformeerd in een punt P', dat op de rechte lijn door M en P ligt en waarvoor de genoemde relatie geldt.
Wanneer P binnen de cirkel ligt, dan ligt P' erbuiten en omgekeerd. Wanneer P op de cirkel ligt, dan valt P' met P samen en is dus een dekpunt. De inversie doet veel denken aan een lijnspiegeling, waarbij een as van spiegeling bestaat waarvan alle punten dekpunten zijn. Wanneer de inversie twee keer achter elkaar uitgevoerd wordt, zijn alle punten weer op hun oorspronkelijke plaats teruggekeerd, net als bij de lijnspiegeling en ook bij de puntspiegeling.
De meetkundige inversie heeft een aantal prettige eigenschappen die met een beetje meetkunde

eenvoudig te bewijzen zijn, maar hier laten we het alleen bij hun vermelding. De belangrijkste eigenschap is wel dat cirkels in het algemeen weer in cirkels getransformeerd worden. Dat 'in het algemeen' moet zo verstaan worden, dat in uitzonderingsgevallen een cirkel overgaat in een rechte lijn. Omgekeerd wordt een rechte lijn of weer in een rechte lijn, of in een cirkel getransformeerd. Bij deze eigenschap zijn we geneigd rechte lijnen te interpreteren als limietgevallen van cirkels, als cirkels met een oneindig grote straal. We kunnen in concrete gevallen gemakkelijk voorspellen wat er bij transformatie zal gebeuren. Het middelpunt M van de inversiecirkel verdwijnt bij transformatie in het oneindige. Wanneer we dus een cirkel willen inverteren welke door dat middelpunt gaat, dan zal de beeldfiguur door het oneindig verre punt gaan, zoals we dat plegen te zeggen. Dan kan het beeld geen gewone cirkel meer zijn en is het dus een rechte lijn. Omgekeerd is het beeld van een rechte lijn die niet door M gaat een cirkel die wel door M gaat. De tweede belangrijke eigenschap is dat de inversie de grootte van hoeken niet aantast. De hoek welke twee cirkelbogen of rechte lijnen in een snijpunt met elkaar maken wordt door de inversie niet veranderd. In figuur 3.21 demonstreren we die eigenschap voor de transformatie van een deel van een schaakbord. De vierkante velden gaan over in door cirkelbogen begrensde vlakjes, waarbij de bogen elkaar loodrecht doorsnijden.



Figuur 3.21 De inversie van een schaakbordfragment



Figuur 3.22 De inversie van de voorplaat

Het geheim van de voorplaat

De fractal op de omslag van dit boek is met behulp van inversie tot stand gekomen. In feite is het niets anders dan een versie van de bekende Mandelbrot-verzameling. De meest bekende vorm ziet eruit als een combinatie van een niervormig gedeelte en een cirkel. Daaromheen bevindt zich een groot aantal kleine schijfjes van steeds kleinere afmetingen. Ten slotte gaat alles over in een ingewikkelde rand, die slechts bij een behoorlijke vergroting zijn volle schoonheid toont. Een wat minder bekende vorm–bij de functie $cz(1-z)$ behorend–is links-rechts symmetrisch en bestaat in hoofdzaak uit twee zuivere cirkels met daaromheen de bekende versierselen. Welnu, een van die cirkels is als cirkel van inversie gekozen. De tweede cirkel, die uitwendig aan de inversiecirkel raakt, gaat door de transformatie over in een kleinere cirkel, die de inversiecirkel nu inwendig raakt. In figuur 3.22 kunnen we zien hoe dat werkt.

In computerprogramma's hebben we formules nodig. Het eenvoudigst zijn die van een inversie waarvan de inversiecirkel in de oorsprong gecentreerd is. Dan is

(3.10)

$x' = r^2 x / (x^2 + y^2)$

$y' = r^2 y / (x^2 + y^2)$

Een voorbeeld

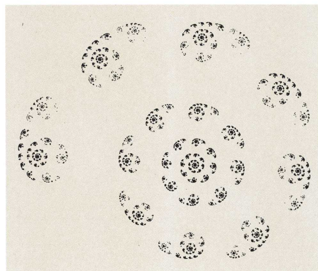
De meetkundige inversie kan heel goed als bouwsteen gebruikt worden voor het samenstellen van fractals. Weliswaar is de inversie geen contractie, maar een slimme combinatie van een inversie en een contractie kan heel werkzaam zijn. Bij wijze van voorbeeld beschouwen we de inversie bepaald door de eenheidscirkel en een draaivermenigvuldiging met dekpunt (2,0), een rotatiehoek van 45 graden en een contractiefactor 0.9. De formules die de kern van het programma vormen, zijn

$x' = x / (x^2 + y^2) \quad y' = y / (x^2 + y^2)$

en

$x' = ax - by + (2 - 2a) \quad y' = bx + ay - 2b$

Met het programma INVERMC verkrijgen we dan de in figuur 3.23 gegeven fractal.



Figuur 3.23 Een fractal als resultaat van een inversie en een draaivermenigvuldiging

Dynamische systemen

De afgelopen eeuwen zijn gekenmerkt door een voortdurende mechanisering van het wereldbeeld, een ontwikkeling die we kunnen laten beginnen bij Archimedes, die zijn hoogtepunt beleeft bij Newton en zijn vervolmaking in de afgelopen honderd jaar. De wetten die de beweging van de hemellichamen beheersen, blijken zo nauwkeurig te zijn dat ze ons, geholpen door de rekenmachines, in staat stellen een kosmische verspieder naar de uithoeken van het zonnestelsel te sturen, precies daarheen waar zich interessante objecten bevinden. Dynamica is de leer van de krachten, en het zonnestelsel vormt dan ook het prototype van een dynamisch systeem waarbij de zwaartekracht de uitverkoren kracht is die alles in beweging brengt. Geen wonder dat we denken dat we op soortgelijke wijze wel meer kunnen berekenen. Het behoort al eeuwen lang tot een goede traditie om zons- en maansverduisteringen lang van tevoren nauwkeurig te kunnen voorspellen. We zouden nu toch wel in staat moeten zijn met de snelle rekenapparatuur op zijn minst het weer van de volgende dag nauwkeurig te kunnen voorspellen. De ervaring wijst uit dat de resultaten tot dusverre nogal matig zijn, en eigenlijk gaat het met het moderne computergeweld maar een klein beetje beter dan wat een ervaren boer ons weet te vertellen wanneer hij naar de wolken kijkt en zijn vinger in de lucht houdt.

Nu is meteorologie een vrij ingewikkelde zaak, waarbij een enorm aantal elementaire lucht- en waterdeeltjes betrokken zijn en waarbij meer krachten in het spel zijn. Laten we ons maar liever eerst bezighouden met een veel eenvoudiger dynamisch probleem, het zogenaamde drielichamenprobleem van Poincaré. Newton kon al bewijzen dat twee elkaar aantrekkende hemellichamen, geïdealiseerd als puntmassa's, elliptische banen beschreven om een gemeenschappelijk zwaartepunt. Poincaré wilde een stapje verdergaan door er nog een derde lichaam aan toe te voegen, maar kwam al ras in grote theoretische problemen.

Nu weten we dat die problemen van principiële aard zijn, en dat zelfs bij sterk vereenvoudigde situaties van dat probleem er zich op den duur een onvoorspelbaar, chaotisch gedrag gaat manifesteren. Men kan zich afvragen hoe dit te rijmen is met de wetten van Newton. Men kan in dit geval het dynamische systeem vatten in de vorm van een wiskundig model, dat uit een paar differentiaalvergelijkingen bestaat. Volgens de theorie legt een gegeven beginsituatie de toekomst volledig vast, ook al zouden we in de tijd naar het oneindige willen gaan. En toch nemen we bij vele dynamische systemen als hier op den duur onvoorspelbaarheid en chaotisch ge-

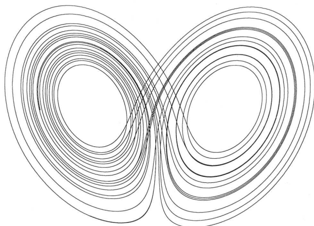
drag waar. Pas sinds kort zijn we gaan begrijpen hoe dat kan. Eenvoudig gezegd komt het erop neer dat we bij de wiskundige berekening voortdurend nauwkeurigheid verliezen. Zouden we over een computer beschikken die getallen hanteert met een nauwkeurigheid van zeg dertig decimalen, dan zouden we daarvan op den duur niets meer overhouden. Dat ligt niet aan de computer, maar aan de aard van het dynamische systeem. We kunnen dat verschijnsel aan de hand van heel eenvoudige voorbeelden goed toelichten, maar eerst willen we iets zeggen over de beroemde experimenten van de meteoroloog Edward Lorenz, waarover ook elders uitvoerig is geschreven.

De vlinder van Lorenz

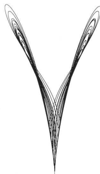
In 1961 maakte Lorenz computerberekeningen van een meteorologisch model bestaande uit twaalf differentiaalvergelijkingen. Het viel hem op dat de verkregen grafiek sterk afhing van de gebruikte numerieke nauwkeurigheid, en hij vroeg zich terecht af of het resultaat op langere tijdschaal nog wel realistisch was. Hij had sterk de indruk dat een heel kleine verandering aan het begin de berekening op den duur ernstig verstoorde, en hij gaf uiting aan zijn twijfel in de titel van een van zijn publikaties: 'Kan het fladderen van een vlinder in Brazilië een orkaan doen losbarsten in Texas?' Later bleek dat hij gelijk had en dat het door hem ontdekte effect, nu het 'vlindereffect' genoemd, heel veel voorkomt. In 1963 beschreef Lorenz een sterk vereenvoudigd wiskundig model dat praktisch dezelfde eigenaardigheden vertoonde als van het eerste, ingewikkelder model. Het is het volgende stelsel van drie differentiaalvergelijkingen:

(4.1)
$$dx/dt = -\sigma x + \sigma y$$
$$dy/dt = rx - y - rz$$
$$dz/dt = -bz + xy$$

waarbij t de tijd beschrijft. Lorenz gaf aan σ de waarde 10, aan b de waarde 8/3 en ging experimenteren met aan r verschillende waarden toe te kennen. Een oplossing van dat stelsel gaf hij op het beeldscherm van zijn computer weer als een ruimtelijke kromme, uiteraard in de vorm van een projectie. Volgens de wiskundige theorie is door de situatie op een begintijdstip $t = 0$ de oplossing ondubbelzinnig vastgelegd. In meetkundige termen betekent dit dat de oplossingskromme volledig bepaald is door alleen zijn beginpunt. Lorenz nam onder andere de waarde $r = 28$ en vond dat er een heel eigenaardige, uit vele lussen bestaande figuur gevormd werd. Nam hij telkens een ander beginpunt, dan vond hij op den duur toch weer dezelfde lussen. Het leek wel of er zich in de ruimte een eigenaardig object bevond waar alles heenging, een soort 'attractor'. Nam hij een ander beginpunt, dan bleek de gevormde kromme na een paar slingers zich al praktisch op de 'attractor' te bevinden. De merkwaardige, door Lorenz gevonden figuur is het eerste voorbeeld van een zogenaamde 'strange attractor' of 'vreemde aantrekker'. In de figuren 4.1 en 4.2 tonen we twee projecties. De aantrekker lijkt te bestaan uit een oneindig aantal lussen



Figuur 4.1 Frontale projectie van de Lorenz-aantrekker



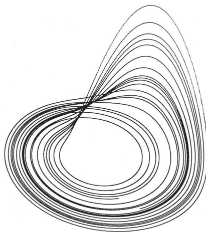
Figuur 4.2 Zijdelingse projectie van de Lorenz-aantrekker

die zich als het ware op de pagina's van een half-opengeslagen boek bevinden.

Deze illustraties kan men met weinig moeite zelf op het beeldscherm krijgen door gebruik te maken van het programma LORENZ1. Met het vereenvoudigde model van Lorenz kan men ook het vlindereffect op fraaie wijze illustreren. In het programma LORENZ2 gaan we uit van een aantal zeer dicht bij elkaar gelegen beginpunten. Elk van die punten genereert een oplossingskromme, een baan. Die baan tekenen we niet, maar wel markeren we telkens de ogenblikkelijke positie van elk van die punten op opeenvolgende tijdstippen. Het is alsof er een microscopisch treintje vertrekt met een flink aantal wagons. Wanneer we een snelle computer hebben en we ook nog gebruik maken van een gecompileerde versie van Basic, dan zien we voor onze ogen zich een film ontrollen. Het treintje volgt globaal een aantal lussen van de Lorenz-aantrekker, maar wordt onderwijl telkens iets langer. De losse punten die eerst nog een klein lijnstukje leken te vormen, verliezen hun samenhang en ten slotte zien we een uitgerekte rij van puntjes over het scherm bewegen. Op een bepaald moment komt er een splitsing, de ene helft gaat naar links, de andere naar rechts. De trein wordt meer en meer gedesintegreerd, totdat er een wolk punten zich ordeloos, chaotisch zeggen we, over het scherm beweegt. De punten bevinden zich weliswaar min of meer precies op de aantrekker, maar het is onvoorspelbaar wáár ze zich bevinden.

Meer vreemde aantrekkers

In de afgelopen jaren is wat we nu de 'Lorenz-attractor' noemen uitvoerig bestudeerd en er zijn vele andere soortgelijke objecten gevonden. In de regel zijn het fractals en hebben ze een gebroken dimensie. In een situatie als die van het Lorenz-model is de aantrekker een soort stabiele, lijnvor-



Figuur 4.3 Een Roessler-aantrekker

mige figuur. Stel eens dat we in plaats van slechts een treintje van enkele punten alle punten van een grote bol als beginpunt nemen, waarbij we de bol zo groot nemen dat de Lorenz-aantrekker er royaal in past. We zien dan dat de bol zich in de loop van de tijd vervormt, waarbij in twee richtingen verkorting en in de derde richting verlenging plaatsvindt. De bol wordt eerst een soort worst, dan een knoedeltje vermicelli en ten slotte een dunne draad met vele lussen. Het uiteindelijke lot van de massieve bol is een oneindig lange kronkellijn, de aantrekker.
In 1979 gaf Otto Roessler van de Universiteit van Stuttgart nog een paar voorbeelden van vreemde aantrekkers. Zijn modellen betreffen ook weer een stelsel van drie differentiaalvergelijkingen en evenals bij Lorenz kan het oplossen van een dergelijk stelsel geïnterpreteerd worden als het vormen van een ruimtelijke kromme.
Een van zijn modellen vermelden we hieronder:

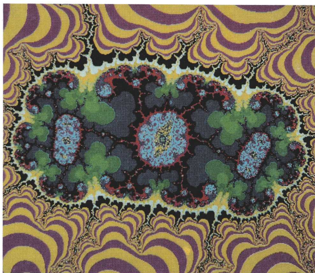
(4.2)
$dx/dt = -y - z$
$dy/dt = x + ay$
$dz/dt = bx - cx - xz$

waarbij a, b en c bepaalde welgekozen getallen zijn.
Onder bepaalde omstandigheden gaan alle oplossingen weer naar een zelfde limietstructuur toe, een zogenaamde Roessler-aantrekker, die in figuur 4.3 is afgebeeld.
Met een programma van het type ROESSLER kan men deze illustratie en soortgelijke figuren in een willekeurige projectie op het scherm krijgen.

Iteratieve dynamische systemen

Het systeem van Lorenz is een zogenaamd continu dynamisch systeem, omdat de tijd zonder onderbreking doorloopt. Een dergelijk systeem is meestal gekenmerkt door een beschrijving met differentiaalvergelijkingen. De wetten van Newton zijn geformuleerd als differentiaalvergelijkingen en in die zin zijn praktisch alle mechanische systemen, van een eenvoudige slinger tot ingewikkelde uit vele delen bestaande kosmische stelsels toe, te beschouwen als continue dynamische systemen. Wanneer we echter dergelijke wiskundige modellen voor de computer geschikt willen maken, zijn we wel gedwongen de tijd in kleine stukjes te hakken. Differentialen moeten vervangen worden door differenties, de deftige naam voor kleine verschillen. Er heeft zich een heel vakgebied – de numerieke wiskunde – ontwikkeld, waarin op geraffineerde wijze de differentiaalvergelijkingen van

Een bacil van Mandelbrot (vergrotingsfactor 1.5 × 10⁹)



Newton zo goed mogelijk geïmiteerd worden. Dankzij deze discipline konden de bewegingen van ruimtevaartuigen zo uiterst precies berekend worden.

In het algemeen spreken we van een *iteratief dynamisch systeem* wanneer de tijd met stapjes voortschrijdt, wanneer het systeem als het ware telkens een klein beetje, maar toch schoksgewijs, verandert. We kunnen constateren dat een continu dynamisch systeem een object is voor alleen theoretische studie, de theoretische mechanica, maar dat we het dienen te vervangen door een iteratief dynamisch systeem zodra er met de computer aan gerekend moet worden. De term 'iteratief' (van een Latijnse stam, die herhalen betekent) betekent dat we telkens van een willekeurig tijdstip tot het volgende tijdstip dezelfde berekening moeten uitvoeren, almaar een herhaling van dezelfde berekening met dezelfde formules.

Bij sommige toepassingen geeft een iteratief dynamisch systeem een veel passender beschrijving van een natuurlijk proces. Dat is in het bijzonder het geval in de zogenaamde populatiedynamica, dat gebied van de biologie waarin men bijvoorbeeld modellen van geremde groei, of de interactie tussen verschillende diersoorten in hun onderlinge strijd om het bestaan, bestudeert. Voor ons het eenvoudigste model is dat van Verhulst, waarin de geremde groei beschreven wordt van opvolgende generaties, van insekten bijvoorbeeld. Het model is tegelijk zo interessant en belangrijk, dat er al veel over geschreven is op alle niveaus van vlot leesbaar en populair tot hoog-wetenschappelijk en moeilijk toegankelijk.

Het model van Verhulst

Bij dit veelbesproken model beperken we ons hier tot enkele hoofdzaken. Het model wordt beschreven door de iteratieve vergelijking

(4.3)
$x_{n+1} = a x_n (1 - x_n)$

waarbij $0 \leq x_n \leq 1$ en $0 < a \leq 4$.
De biologische betekenis van x_n is het relatieve aantal van de *n*de generatie van een diersoort, waarbij 1 het maximale aantal, de draagkracht van het milieu, aangeeft. De index *n* speelt de rol van een met stapjes verlopende tijd. De parameter *a* is een maat voor de vruchtbaarheid. Wanneer $a < 1$, is de vruchtbaarheid te gering en sterft de diersoort uit. Hoe we werken met een dergelijk model kunnen we het beste aan de hand van een eenvoudig voorbeeld zien. Kiezen we bijvoorbeeld $a = 1.5$ en beginnen we op het tijdstip 0 met de *x*-waarde 0.1, dan kunnen we de volgende waarden aflezen in het tabelletje

0	1	2	3	4	5
0.1	0.1350	0.1752	0.2167	0.2546	0.2847
6	7	8	9	10	...
0.3055	0.3182	0.3254	0.3292	0.3313	

In dit voorbeeld gaat x_n voor $n \to \infty$ naar de limietwaarde 0.3333 of exact 1/3, wat we gemakkelijk kunnen verifiëren door die waarde voor x_n en x_{n+1} te substitueren. In het algemene geval kunnen we op dezelfde wijze vaststellen dat $x = 1 - 1/a$ de overeenkomstige waarde geeft. Wanneer we het proberen voor $a = 2.5$ gaat het goed, maar wanneer we bijvoorbeeld $a = 3.2$ nemen blijkt de rij waarden niet naar een enkele limiet te gaan, maar ten slotte heen en weer te springen tussen twee waarden. Een nauwkeuriger onderzoek laat zien dat er nog veel meer aan de hand is wanneer we *a* langzaam laten aangroeien. Het kan gebeuren dat x_n voor $n \to \infty$ naar een stelsel van *m* punten toegaat, een zogenaamde periodieke limietcyclus, kortweg een *m*-cyclus, anders gezegd een uit *m* punten bestaande aantrekker. Met behulp van het programma VERHULST kan men het zelf met de computer narekenen. De aantrekker kan echter ook uit oneindig veel punten bestaan en dus veel ingewikkelder zijn, te vergelijken met de vreemde aantrekker van Lorenz. Zelfs het vlindereffect, het kenmerk van een vreemde aantrekker, kan men bij dit allereenvoudigste model reeds constateren.

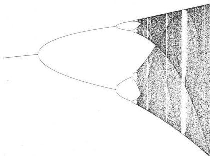
De grote lijnen

Wat nu volgt is van groot belang voor het inzicht in dynamische systemen en de ermee verbonden vreemde aantrekkers en fractals. In het model van Verhulst bestaat voor elke waarde van *a* tussen 0 en 4 een aantrekker en men kan zich gaan afvragen hoe die aantrekker afhangt van *a* wanneer we

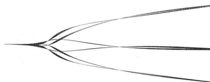
a langzaam laten toenemen vanaf de beginwaarde 0. Het blijkt dat de aantrekker in het algemeen een poosje min of meer ongewijzigd blijft, maar voor een bepaalde waarde van a plotseling van aard verandert. Dat verschijnsel staat bekend als *bifurcatie*, letterlijk vertakking, als bij een vork waar een steel in twee of meer tanden overgaat. Men kan dan een diagram maken waarin op de een of andere manier die bifurcaties in afhankelijkheid van a zichtbaar gemaakt kunnen worden, een zogenaamd bifurcatiediagram. Wat we bij dit allereenvoudigste model van Verhulst aan bijzonderheden kunnen aantreffen geldt analoog bij veel ingewikkelder systemen. De aantrekkende getallenreeksen van dit model kunnen mooie fractals en fraaie Julia sets worden. Het bifurcatieplaatje kan uitgroeien tot de fameuze Mandelbrot set. Het bifurcatiediagram van het Verhulst-model is overigens al ingewikkeld genoeg, het is in wezen niets anders dan een doorsnede langs de symmetrieas van de Mandelbrot set.
Wanneer we een dynamisch systeem willen onderzoeken, dan vragen we in het algemeen naar de aantrekkers. Een aantrekker bepaalt het gedrag van het dynamische systeem op de lange duur, zegt iets over de mogelijke stabiliteit van het systeem. Is een aantrekker slechts een enkel punt, dan is dat vanzelf een evenwicht, vanuit een willekeurig begin gaat het stelsel naar dat punt, naar dat evenwicht toe. Een verstoring van dat evenwicht herstelt dat evenwicht weer. In het algemeen mag een verstoring niet te groot zijn. De betekenis van een evenwicht is doorgaans van lokale aard. Het kan gebeuren dat een dynamisch systeem meer mogelijke evenwichten heeft en dan hangt het helemaal van de beginsituatie af naar welk evenwicht het systeem toegaat. In overdrachtelijke zin rekent men een oneindig ver punt soms als evenwicht, maar het is duidelijk dat bij praktische toepassingen iets als een oneindig verre aantrekker weinig aantrekkelijk is.

Het bifurcatiediagram van het Verhulst-model

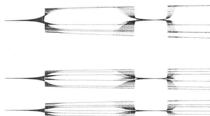
Een deel van het bifurcatiediagram van het model (4.3) van Verhulst is afgebeeld in figuur 4.4. Het diagram is in de literatuur uitvoerig besproken, zodat we er hier kort over kunnen zijn. Met behulp van het programma BIFUR1 kan men het zelf op het beeldscherm van de computer zichtbaar maken. In de illustratie is a horizontaal uitgezet van 0 tot 3.57. Verticaal is de aantrekker, of althans een benadering daarvan, uitgezet. Van links naar rechts gaande zien we hoe de aantrekker aanvankelijk uit een enkel punt bestaat.
Bij $a = 1$ treedt een soort bifurcatie op, maar de aantrekker blijft nog enkelvoudig, totdat bij de waarde $a = 3$ de aantrekker in twee delen splitst. Bij de waarden $a = 3.449490$, $a = 3.544090$, $a = 3.564407$ enzovoort treedt andermaal verdubbeling op en dat gaat door tot aan de waarde $a = 3.669202$. De bifurcatiepunten liggen steeds dichter bij elkaar als de termen van een meetkundige rij. Mitchell Feigenbaum was een der eersten die deze opmerkelijke regelmaat opmerkte en beschreef. Hij vond ook dat dit een heel algemeen verschijnsel was dat op dezelfde wijze bij allerlei iteratieve processen optrad. De betekenis van dit verschijnsel was van groot belang voor de theoretische natuurkunde bij de theorie van overgangsprocessen, zogenaamde faseovergangen zoals smelten en verdampen.



Figuur 4.4 Het bifurcatiediagram van het Verhulst-model



Figuur 4.5 Het bifurcatiediagram voor $2.98 < a < 3.08$



Figuur 4.6 Het bifurcatiediagram voor $3.377 < a < 3.392$

Strijd in de insektenwereld

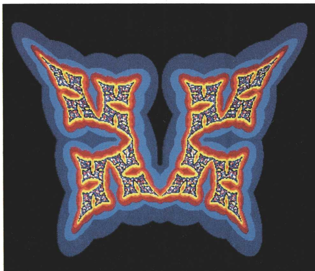
Bij het bovenstaande rekenvoorbeeld hebben we gezien dat er periodieke evenwichten kunnen bestaan. Voor ingewikkelder iteratieve dynamische systemen geldt dat ook, en we vestigen ter illustratie de aandacht op het volgende iteratieve systeem

(4.4)

$$x_{n+1} = a x_n (1 - x_n - y_n)$$
$$y_{n+1} = a x_n \, y_n$$

afkomstig uit de populatiedynamica en verwant aan het bekende model van Verhulst. Het staat nu model voor de wisselwerking tussen twee insektensoorten, te vergelijken met roofdieren en prooidieren, waarbij dus de ene soort leeft ten koste van de andere soort. Dergelijke modellen worden tegenwoordig in de landbouw gehanteerd bij de studie van de biologische bestrijding van schadelijke insektensoorten. Hierbij hebben x en y de betekenis van een relatief aantal individuen en heeft de index n betrekking op de generatie.

Het is niet moeilijk om te verifiëren dat $x = 1/a$, $y = 1 - 2/a$ een evenwicht is, maar het is natuurlijk maar de vraag in hoeverre dat evenwicht stabiel en dus aantrekkend is. Laten we de parameter a toenemen vanaf de waarde 2, dan is het evenwicht alleen maar stabiel tot aan de waarde 3, en daarna gebeuren er vreemde zaken. Experimenteel kan men een indruk van de verschillende mogelijkheden krijgen door een bifurcatiediagram als dat van figuur 4.4 te maken. In de figuren 4.5 en 4.6 hebben we daarvan met behulp van het programma BIFUR2 twee fragmenten afgebeeld. In figuur 4.5 is het traject (2.98,3.08) weergegeven met verticaal de limietwaarden van x_n. In figuur 4.6 is op overeenkomstige wijze het traject (3.377,3.392) afgebeeld. De illustraties spreken voor zich. Dergelijke bifurcatiediagrammen geven al enigszins aan wat we kunnen verwachten bij de Mandelbrot sets, bifurcatiediagrammen waarbij twee parameters betrokken zijn.



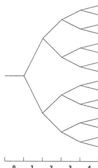
Een zelfgelijkvormige vlinder



Hoogtekaart van een fractaloppervlak gegenereerd door een golfpatroon

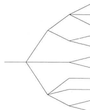
Boomstructuren

In de voorgaande hoofdstukken hebben we al met een aantal boomachtige fractals kennisgemaakt. In dit hoofdstuk kijken we er nog eens wat systematischer naar, en in het bijzonder gaan we in op de praktische aspecten wanneer we een boomstructuur met een computer willen bewerken. Een boomstructuur kunnen we opvatten als een discreet dynamisch proces van elementen, levende cellen bijvoorbeeld, die zich op bepaalde tijdstippen, met de tikken van een imaginaire klok, in delen splitsen. Het schema is als geschetst in figuur 5.1 of figuur 5.2. Het beginelement plegen we de stam te noemen. De elementen van de volgende generaties heten dan takken. In figuur 5.1 zien we een zogenaamde binaire boomstructuur, een structuur welke gekenmerkt is door het getal 2. Van elke generatie verdeelt een tak zich in precies twee (zij)takken. Het aantal takken van de generaties, te beginnen met het rangnummer 0, is gelijk aan 1, 2, 4, 8, 16, 32,... Het aantal takken van de generatie met rangnummer g is dus 2^g, en het totale aantal is dan $2^{g+1} - 1$.



Figuur 5.1 Het schema van een binaire boom

Men realisere zich goed dat figuur 5.1 geen meetkundige betekenis behoeft te hebben. Het is primair bedoeld als een boekhoudkundig schema om de structuur van een boom aan te geven. De lengte van de takken is daarbij niet van belang. Maar natuurlijk kan dezelfde illustratie ook in meetkundige zin gebruikt worden als een weergave van een 'echte' boom. De takken corresponderen dan, afgezien van hun lengte, met echte boomtakken.



Figuur 5.2 Een algemenere boomstructuur

In hoofdstuk 1 hebben we kennisgemaakt met twee meetkundige realisaties van een binaire boomstructuur. Bij de in figuur 1.5 afgebeelde kale boom van Pythagoras corresponderen de takken van de boomstructuur met lijnstukjes, maar bij de in hoofdstuk 1 besproken en in de figuren 2.6 en 2.7 afgebeelde lijn van Lévy worden daarentegen punten afgebeeld.

In de natuur nemen we waar dat bij een echte boom de takken zich niet altijd in twee delen splitsen, maar dat dat aantal soms drie of meer kan zijn, en in het algemeen zelfs variabel kan zijn. Het ligt dus voor de hand het schema van figuur 5.1 te generaliseren en ook een schema als dat van figuur 5.2 te beschouwen, waarbij bijvoorbeeld takken zich maximaal in drieën kunnen splitsen.

Ter wille van de boekhouding, nodig voor het schrijven van een computerprogramma, kan het noodzakelijk zijn de takken van een boom op een overzichtelijke manier te nummeren, zodanig dat we in staat zijn door het aangeven van een of meer rangnummers een bepaalde tak te identificeren. Bij een ingewikkelde boomstructuur met een variabel aantal splitsingen

van een tak zit er niet veel anders op dan een willekeurige tak aan te duiden door een rijtje getallen, zoals bijvoorbeeld 1,3,2,5,2,6,..., te lezen als volgt: na de stam (1) nemen we de derde tak (3) en daarvan weer de tweede zijtak (2) en daarvan weer de vijfde vertakking (5) enzovoort. We tellen dus generatie voor generatie, te beginnen met de generatieindex g, nul voor de stam.

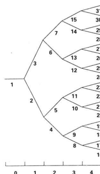
De nummering van een binaire boom

Voor een binaire boom, en trouwens voor elke soortgelijke regelmatige boomstructuur, kan de nummering veel eenvoudiger geschieden. Het totale aantal takken, generatie voor generatie, kunnen we aflezen uit het volgende tabelletje

g	0	1	2	3	4	5	6	...
n	1	2	4	8	16	32	64	...

Wanneer we de takken achter elkaar gewoon doornummeren vanaf 1 als

1; 2,3; 4,5,6,7; 8,9,10,11,12,13,14,15;...



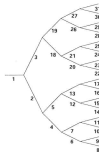
Figuur 5.3 Tellen van takken

dan kunnen we voor een willekeurig rangnummer n snel vaststellen waar de desbetreffende tak zich in de boom bevindt. In figuur 5.3 zien we dat een tak met het rangnummer r zich splitst in twee takken met de rangnummers $2r$ en $2r+1$. Door terug te werken, kunnen we dan precies vaststellen waar een tak met een bepaald rangnummer zich bevindt. Laten we eens naar de tak met rangnummer 43 kijken. Blijkbaar is deze tak een afsplitsing van die met nummer 21, en zo kunnen we telkens door halvering teruggaan tot aan de stam:

$43 \rightarrow 21 \rightarrow 10 \rightarrow 5 \rightarrow 2 \rightarrow 1$

De tak heeft dus het generatiegetal 5. Zijn we alleen in dat generatiegetal geïnteresseerd, dan kunnen we g ook eenvoudig bepalen als de kleinste gehele exponent waarvoor $2^g < n$. In het gegeven voorbeeld is $n = 43$ en $2^5 = 32$, terwijl $2^6 = 64$.

In het computerprogramma TREESB, waarin de fractal van Lévy (figuren 2.6 en 2.7) punt voor punt wordt opgebouwd, is het een ander nader uitgewerkt. Telkens worden de takken van een nieuwe generatie uit die van de voorafgaande afgeleid. Daartoe is het nodig in het computergeheugen – 1 megabyte is al veel – ruimte te reserveren voor de numerieke gegevens van opeenvolgende generaties. Die ruimte kan behoorlijk oplopen wanneer het generatiegetal groter dan tien wordt. Voor de orde $p = 12$, wanneer het aantal takken 8095 bedraagt, kan ook een moderne computer al de mededeling 'out of memory' geven. Wanneer men bescheiden in zijn wensen is, kan men dit programma zien als een goed compromis ten aanzien van eenvoud en effectiviteit. Wil men meer, dan moet er dieper nagedacht worden. Dat dit de moeite loont zal hierna blijken.



Figuur 5.4 Het schema van de backtrackmethode

Het struikelblok bij de vorige methode was het beperkte werkgeheugen van de computer. Laten we daarom even stilstaan bij de vraag hoe we daarvan een beter gebruik kunnen maken. Tot dusverre hebben we van elk berekend punt de gegevens in twee arrays $x(n)$, $y(n)$ bewaard, ongeacht of we die later nog nodig zouden kunnen hebben. Het is duidelijk dat tijdens de berekening onnodig veel geheugenruimte geblokkeerd wordt door reeds verwerkte gegevens. Met de nu te bespreken methode van backtracking (een goed Nederlands equivalent is er niet, of het zou 'terugsporing' moeten zijn) gaat het allemaal veel beter. De methode kan men zien als het op systematische wijze zoeken naar een weg in een doolhof. We volgen een bepaalde route, door bijvoorbeeld bij elke splitsing rechtsaf te gaan en te vermijden dat hetzelfde weggedeelte tweemaal wordt afgelegd. Daarom moeten wij al wandelend bij elke splitsing van wegen een kleine boekhouding verrichten.

We stellen ons een binaire boomstructuur voor als in figuur 5.4, die vijf generaties diep gaat. De nummering van de takken is reeds aangepast aan de nu te bespreken methode. Er zijn in totaal 31 takken. We vatten het schema op als een wegennet en we verplaatsen ons in de positie van een postbode die de taak heeft alle wegen af te lopen vanaf de stam. We beginnen met de route 1, 2, 4, 6, 8 waarbij we bij elke splitsing de rechtertak volgen. Aan het eind van dit gedeelte moeten we wel 'op onze schreden terugkeren' tot aan de laatste splitsing. In het Engels spreken we daarbij van 'backtracking', en die technische term nemen we voor deze methode over. De volgende tak is dus nummer 9, maar dan moeten we weer terug en wel tot de eerste splitsing die we nog niet benut hebben. We krijgen nu de nummers 7, 10, 11 waarna we weer een heel stuk terug moeten. Het volgende gedeelte is 5, 12, 14, 15, 13, 16, 17 en dan weer een flink stuk terug tot waar de stam zich splitst. De rest gaat op dezelfde manier.

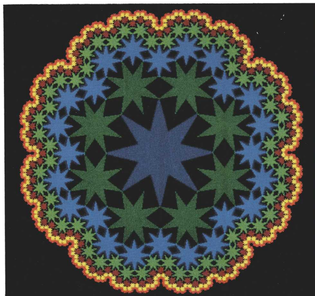
De methode is verwerkt in het programma TREEBT, dat we bespreken aan de hand van figuur 5.4. In dit programma wordt een heel zuinig gebruik gemaakt van het computergeheugen, en de enige praktische beperking is slechts de tijdfactor. Wie een snelle computer heeft, kan in minder tijd veel meer generaties berekenen, maar het hangt er maar van af of dat werkelijk zin heeft. Essentieel voor deze methode is dat van tevoren een uitspraak gedaan wordt over het aantal te berekenen generaties. Dat aantal, de zogenaamde orde, duiden we in het programma doorgaans aan met de letter p. Uiteraard is het programma geschreven voor willekeurige waarden van p en moet dus bij elke concrete toepassing aan het begin voor p een waarde gekozen worden. We kiezen ter illustratie de waarde $p = 4$. Er zijn dan in totaal $2^5 - 1 = 31$ takken, waarbij er voor de af te beelden punten evenzovele coördinaten x,y bepaald moeten worden. Het programma berust in wezen op het begrip 'stapel'. Berekende gegevens die we later nog een keer moeten gebruiken, bergen we op in een array dat we ons kunnen voorstellen als een stapeltje systeemkaarten. Nieuwe kaarten komen bovenop de stapel terecht en benodigde gegevens worden boven van de stapel gehaald volgens het beginsel 'last in first out'.

Het eerste punt, in het programma het punt (0,1), is onze start. Daarna

Een Julia set bepaald door een tangens functie



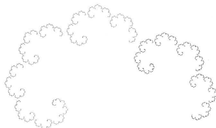
Van ster tot fractal



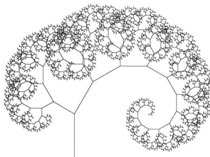
worden de punten met de rangnummers 2 en 3 berekend. Met 2 gaan we verder, maar de coördinaten van 3 bergen we voor later gebruik op als $x(1)$ en $y(1)$. Daarbij houden we in een array de generatie bij als $z(1)$. Het punt 1 hoeven we nu niet meer te gebruiken. Uit 2 leiden we de punten met de rangnummers 4 en 5 af. Met de rechtertak gaan we verder, maar de gegevens van de linkertak bergen we weer op als $x(2)$, $y(2)$, $z(2)$. Bij de volgende stap gaan we verder met punt 6 en hebben we de gegevens van punt 7 als $x(3)$ enzovoort bovenop de stapel gelegd. Nu aangekomen bij de voorlaatste tak gaat er iets anders gebeuren. De gegevens van de punten met rangnummer 8 en 9 worden berekend en de punten worden afgebeeld. Er behoeft nu niets meer opgeborgen te worden, maar wel moet, om verder te kunnen gaan, de bovenste kaart van de stapel gehaald worden. Dat betreft punt 7 en nu kunnen ook de gegevens van de takken 10 en 11 berekend en verwerkt worden. Om verder te kunnen gaan, kijken we weer naar de bovenste kaart van de stapel waar we de gegevens van punt 5 vinden. We gaan op dezelfde wijze steeds verder tot we niet meer kunnen, en dat merken we doordat de stapel uitgeput is.
Het beschreven programma is een soort prototype van dergelijke programma's. Omdat van elk af te beelden element het generatienummer steeds ter beschikking staat, leent het programma zich ook voor toepassing van kleuren waarbij elke generatie zijn eigen kleuren krijgt. Het is ook niet moeilijk een soortgelijk programma voor een boom met een drietallige structuur te schrijven. Het programma SIERBT geeft daarvan een voorbeeld. Daarin wordt uit een drietal centrale vermenigvuldigingen een stoffractal opgebouwd waarvan de Sierpinski-zeef een speciaal geval is.

Voorwaardelijke terugkeer

Bij het bovenstaande schema van backtracking gaan we uit van een boomstructuur waarbij het aantal te berekenen en te tekenen generaties, in het programma aangeduid met p, van tevoren vaststaat. Stelselmatig bewegen we ons in de boom in voorwaartse richting, voorzover mogelijk de rechtertak kiezende, totdat de laatste generatie aan de beurt komt. Dat geeft het signaal om terug te keren naar takken die nog niet aan de beurt geweest zijn. Er is echter ook een heel ander principe van terugkeer mogelijk, namelijk door te letten op de mate van verkorting. We nemen aan dat elke volgende linkertak met de factor f_1 verkort is, terwijl rechtertakken de reductiefactor f_2 bezitten. Het is niet moeilijk in het programma bij de berekening van de positie van een willekeurige tak tegelijk de bijbehorende reductiefactor te bepalen. Beschikken we over de gegevens van een willekeurige tak en is f daarvan de reductiefactor, dan zijn ff_1 en ff_2 de overeenkomstige factoren van de volgende takken. We spreken nu af dat we terugkeren in het schema van backtracking zodra de totale verkorting beneden een van tevoren bepaalde drempel is gekomen.
Met maar heel weinig meer kunnen we de twee principes van terugkeer combineren. In het programma TREEBTA is dit geschied, en keren we 'op onze schreden terug' zodra of de laatste generatie p bereikt is, of zodra de verkorting beneden de gegeven drempel komt. Het resultaat van dat programma, een spiraalvormige puntenwolk, is afgebeeld in figuur 5.5.



Figuur 5.5 Een spiraalvormige puntenwolk



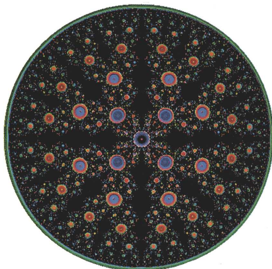
Figuur 5.6 Een asymmetrische kale boom van Pythagoras

Een scheve boom van Pythagoras

In figuur 5.6 hebben we een zogenaamde scheve boom van Pythagoras afgebeeld, welke ten duidelijkste zelfgelijkvormig is. Die boom is gebaseerd op twee contracties, draaivermenigvuldigingen waarbij niet alleen de rotatiehoeken, maar ook de contractiefactoren verschillen. Wie benieuwd is naar het aantal getekende generaties kan terugtellen vanaf het kleinste takje tot aan de stam. Maar reeds een oppervlakkige beschouwing laat zien dat dat aantal sterk uiteen kan lopen. Het is niet moeilijk er enige theorie aan te verbinden, maar liever geven we een concreet voorbeeld waarbij de reductiefactor van de linkertak 1/2 en die van de rechtertak 1/4 is. Na vier generatiestappen zijn er uit de stam zestien takken ontstaan, waarbij de meest linkse verkort is met de factor 16, terwijl de meest rechtse tak een factor 256 verkleind is. In de figuur is de laatste tak praktisch tot een punt gereduceerd en heeft het geen zin meer verdere generaties te berekenen. Maar met de meest linkse tak kunnen we nog wel even doorgaan. Om een evenwichtige illustratie te maken, kunnen we afspreken dat we takken kleiner dan een van tevoren afgesproken lengte, de tolerantie of drempel, niet meer afbeelden. In het proces van backtracking kunnen we als in het programma TREEBTA die afspraak inbouwen door in de boomstructuur net zo lang voorwaarts te gaan, totdat de afgesproken drempel bereikt is. In het programma PYT3BT is dat beginsel gebruikt om figuur 5.6 te vormen.

De Monte Carlo-methode

Heinz-Otto Peitgen, een der eersten die de wereld verraste met de vele fraaie fractals die hij uit zijn computer getoverd had, noemt het een 'chaos game', een spel van chaos. Michael Barnsley spreekt in zijn al eerder genoemde boek over een IFS, een 'iteratief functie systeem'. Zij wekken de indruk dat fractals enigszins het toevallige bijprodukt zijn van 'chaos', maar eerder is het tegendeel waar. Een fractal moge er bij oppervlakkige beschouwing enigszins 'rommelig' uitzien, in de regel heeft een fractal juist een zeer regelmatige structuur. In het bijzonder is dat het geval voor wiskundige fractals, waaraan een zelfgelijkvormigheid ten grondslag ligt. Een ideale fractal bestaat uit onaftelbaar veel punten en kan als zodanig nimmer in alle details afgebeeld worden. Wat we wel kunnen doen is een fractal zo goed mogelijk benaderen door een puntenwolk van enige duizenden punten. Dat gaat in vele gevallen heel goed door als het ware op een chaotische manier door de fractal heen en weer te springen. We lichten dit toe aan de hand van een bekend geval.



Een fractalwindroos

Fractaltennis

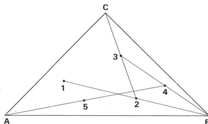
We stellen ons een soort wiskundig tennis voor, dat door drie personen A, B en C met een puntvormige bal op het computerscherm wordt gespeeld. De regels zijn heel strikt. Op het scherm bevinden zich drie punten in overigens een willekeurige positie. Die punten noemen we eveneens A, B en C. Laten we aannemen dat in de loop van het spel de bal zich in de positie van het punt P bevindt en dat persoon A aan slag is. Hij slaat de bal op een zodanige wijze, dat deze terechtkomt in het midden van het lijnstuk AP. Op overeenkomstige wijze zouden B en C aan slag zijnde de bal plaatsen in de middens van de lijnstukken PB en PC. Wie aan de beurt is, wordt door het toeval bepaald. We kunnen ons voorstellen dat de scheidsrechter met een gewone dobbelsteen telkens vaststelt wie aan beurt is met een slag. Natuurlijk moeten we ergens beginnen en dat doet de scheidsrechter heel eenvoudig door de bal zomaar ergens neer te werpen. Bij dit computertennis blijft bij elke slag de vorige positie van de bal bewaard, zodat we na zeg zesduizend slagen een volledig overzicht hebben van het verloop van het spel. Weliswaar is de volgorde van de slagen verloren gegaan, maar goed is te zien waar de bal zoal geweest is.

In figuur 6.1 geven we een beeld van de eerste paar slagen.

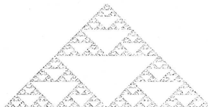
Hoewel het spel desnoods met potlood en papier gespeeld kan worden, kunnen we beter de computer te hulp roepen. In het computerprogramma TENNIS hebben we de tenniswedstrijd gesimuleerd. De kern van dat programma kunnen we in pseudocode als volgt beschrijven:

```
RANDOMIZE (kies een random number generator)
Q1 = 1/3 : Q2 = 2/3 (waarschijnlijkheidsgrenzen)
P = P0 (kies een beginpunt van de baan)
FOR N = 0 TO 6000 (begin van de lus)
R = RND (bepaal een random number)
IF R < Q1 THEN GOSUB A
IF R ≥ Q1 AND R < Q2 THEN GOSUB B
IF R ≥ Q2 THEN GOSUB C
DRAW P (beeld nieuwe baanpunt af)
NEXT N (herhaal de lus)
END
```

Het resultaat van het spel tonen we in figuur 6.2. Eerlijkheidshalve bekennen we dat we daarbij aan het begin een aantal slagen hebben weggelaten.



Figuur 6.1 Het begin van het fractalspel



Figuur 6.2 Na de wedstrijd

Wellicht tot veler verrassing is het resultaat een variant van de bekende zeef van Sierpinski. Maar wie even nadenkt, kon dat wel verwachten. De actie van speler A is immers niets anders dan het uitvoeren van een contraherende afbeelding, een contractie met factor 1/2, in dit speciale geval een centrale vermenigvuldiging met factor 1/2 ten opzichte van het hoekpunt A. Iets overeenkomstigs geldt voor de andere twee spelers. De limietfiguur is dus zelfgelijkvormig ten opzichte van elk van deze drie contracties en ten opzichte van alle combinaties van deze drie basistransformaties. In hoofdstuk 3 hebben we uitvoerig over dergelijke situaties gesproken, en we weten dat voor een gelijkzijdige driehoek ABC het resultaat de Sierpinski-zeef is. Wanneer A, B en C een willekeurige positie hebben, is de ontstane fractal niets anders dan een (affiene) vervorming van de regelmatige Sierpinski-fractal.
In hoofdstuk 3 hebben we alleen gelet op gelijkvormigheidstransformaties, maar nu blijkt aan de hand van het fractaltennisspel op wat voor eenvoudige wijze daarop een fractal gebaseerd kan worden. Er is niets anders voor nodig dan al dobbelend de beschikbare basistransformaties in een willekeurige volgorde toe te passen. De enige voorwaarde is dat de basistransformaties contracties zijn, dat ze dus lijnstukken verkleinen. Dat het beginpunt er helemaal niet toe doet, blijkt gemakkelijk uit de eigenschap van voortdurende contractie. Zouden we de puntenrij beginnen met Q in plaats van met P en passen we in beide gevallen dezelfde rij transformaties toe, dan is na een aantal stappen de lengte van PQ dusdanig verkleind, dat de rij van Q-punten niet meer te onderscheiden is van de rij van P-punten.
In hoofdstuk 4 hebben we over dynamische systemen gesproken, en we hebben laten zien dat een puntenrij gevormd door een iteratieve afbeelding als

$$P \rightarrow \mathrm{T}(P)$$

een aantrekkende limietverzameling kan hebben. In ons geval is T een contractie, gekozen uit een aantal mogelijkheden, en is de limietverzameling een zelfgelijkvormige fractal. De puntenrij convergeert in de regel vrij snel tot de fractal. In de praktijk mogen we ervan uitgaan dat de rij spoedig zo dicht tegen de fractal aanligt, dat we mogen zeggen dat hij er dan deel van uitmaakt. Wie het onderste uit de kan wil hebben, moet er natuurlijk voor zorgen dat het beginpunt van de puntenrij reeds tot de fractal behoort. Dat is niet moeilijk, want de dekpunten van de basistransformaties behoren met zekerheid tot de fractal. Zodra we maar een enkel punt van de fractal kennen, kunnen we nog veel meer punten ervan construeren, eenvoudig door er de basistransformaties op los te laten. In het geval van het tennisspel met de spelregels van Sierpinski behoren de hoekpunten A, B en C tot de fractal, en vervolgens ook de middens van de zijden van de driehoek.
In hoofdstuk 5 hebben we boomstructuren bekeken. We realiseren ons dat het tennisspel beschouwd kan worden als een boom waarbij elke tak zich in drie gelijke delen splitst. Het Monte Carlo-principe betekent dat we in

die boom omhoog klimmen en dat we bij elke splitsing een willekeurige keuze maken. Het lijkt wel een groot wonder dat we al dobbelend toch de hele fractal kunnen bereiken, terwijl we grote gedeelten van de boom onbenut laten. Dat de Monte Carlo-methode zoveel succes heeft, komt dan ook door twee effecten. In de eerste plaats hebben we steeds te maken met contracties, waarbij alles wat eens groot en ver weg is op den duur klein en nabij wordt. In de tweede plaats hebben we te maken met de fundamentele wetten van de waarschijnlijkheidsrekening. Neem bijvoorbeeld de bekende RND(random)-functie van de computer. Dat is steeds een getal tussen 0 en 1, en dat kunnen we afbeelden als een punt op een lijnstuk. Dobbelen we aldus vele honderden punten bij elkaar, dan blijken die het lijnstuk gelijkmatig te vullen. Hoe meer punten we nemen, des te beter gaat het. In de praktijk hebben we voor een goede, gelijkmatige vulling van een tweedimensionale fractal veel meer punten nodig, enige tienduizenden bijvoorbeeld.

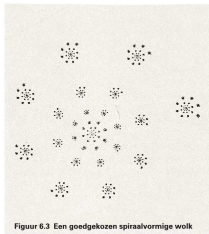
Met meer en met minder kans

In het vorige hoofdstuk hebben we gezien welke maatregelen we moesten nemen om van bijvoorbeeld een scheve Pythagoras-boom toch een evenwichtige afbeelding te krijgen. We gebruikten toen bij de methode van backtracking het criterium van het kleinste te tekenen element, de zogenaamde tolerantie. Bij de Monte Carlo-methode hebben we met een soortgelijk probleem te maken. We beschouwen, zoals in het programma CONTRMC1, een tweetal gelijkvormigheidstransformaties en we volgen daarbij enigszins de opzet ven het programma TREEBTA. De transformatie L heeft de reductiefactor f_1 en de tweede transformatie R heeft de reductiefactor f_2. We kunnen voor f_1 denken aan een waarde die dicht bij 1 ligt, 0.8 bijvoorbeeld, en bij f_2 aan een veel kleinere waarde, zoals 0.1. Wanneer we bij een willekeurige rij van transformaties toch tot een gemiddeld gelijkmatige verkorting willen komen, zal L veel vaker dan R gekozen moeten worden. Omdat $(0.8)^9$ ongeveer gelijk is aan 0.11, moeten we globaal L negenmaal zo vaak uitvoeren als R. In het stochastische proces kunnen we hieraan tegemoetkomen door het invoeren van een waarschijnlijkheidsdrempel q. De kern van het programma is dan

```
FOR N = 0 TO NMAX
R = RND
IF R < Q THEN GOSUB L ELSE GOSUB R
DRAW P
NEXT N
```

Er rest nog de nadere vaststelling van de drempel q. Volgens het programma worden de transformaties L en R respectievelijk met de kansen q en $1-q$ gekozen. Om de gewenste gemiddelde verkorting te bereiken, moeten die kansen zich verhouden als de bijbehorende contracties, dus

$$q:(1-q)=f_1:f_2$$


Figuur 6.3 Een goedgekozen spiraalvormige wolk

Een kleine berekening leert dan dat

$$q = f_1 / (f_1 + f_2)$$

Het hierop gebaseerde programma CONTRMC1 is daarmee bijzonder effectief. Een enkel resultaat is afgebeeld in figuur 6.3.
Men lette erop dat hierbij gebruik is gemaakt van een reductiefactor die op lijnstukken betrekking heeft, de zogenaamde lineaire reductiefactor. Bij een meer 'gevulde' fractal kan het echter beter zijn met de reductie van oppervlakten rekening te houden. Wanneer f de lineaire reductiefactor van een gelijkvormigheidstransformatie is, geeft f^2 de reductie van de oppervlakten. In de praktijk is het daarom verstandig bij de keuze van waarschijnlijkheidsdrempels formules als hierboven alleen als leidraad te beschouwen en zich vooral te laten leiden door experimenten.

De varen van Barnsley

Een fraaie illustratie van de besproken techniek is het in figuur 3.4 afgebeelde varenblad van Michael Barnsley. Aan de figuur liggen vier affiene transformaties ten grondslag van de vorm (zie 3.1)

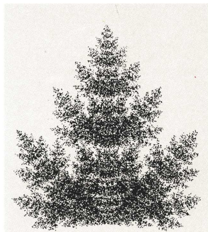
$$x' = ax + by + e$$
$$y' = cx + dy + f$$

De coëfficiënten kunnen verenigd worden in het onderstaande tabelletje

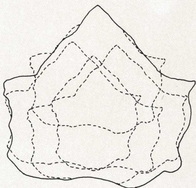
a	b	c	d	e	f	reductie	kans
0	0	0	0.16	0	0	0	0.01
0.85	0.04	−0.04	0.85	0	1.6	0.72	0.85
0.2	−0.26	0.23	0.22	0	1.6	0.10	0.07
−1.5	0.28	0.26	0.24	0	0.44	0.11	0.07

De getallen zijn aldus door Barnsley gegeven. De reductiefactor, hier betrekking hebbend op de oppervlakten, is voor een affiene transformatie gelijk aan $ad - bc$. De kansdrempels zijn hieraan min of meer evenredig.
Van de vele door Barnsley behandelde interessante figuren vermelden we nog het esdoornblad als geïllustreerd in figuur 6.4.
Ook daarbij zijn vier affiene transformaties, te weten twee centrale vermenigvuldigingen en twee krimpdraaiingen. Het programma MAPLE is slechts een variant van het varenprogramma. De tabel is

a	*b*	*c*	*d*	*e*	*f*	reductie	kans
0.6	0	0	0.6	0.18	0.36	0.36	0.3
0.6	0	0	0.6	0.18	0.12	0.36	0.3
0.4	0.3	−0.3	0.4	0.27	0.36	0.25	0.2
0.4	−0.3	0.3	0.4	0.27	0.09	0.25	0.2



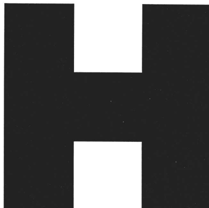
Figuur 6.4 Het esdoornblad van Barnsley



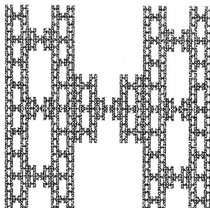
Figuur 6.5 De collage van een esdoornblad

Fractals op bestelling

Michael Barnsley vroeg zich af hoe hij een gegeven meetkundige figuur, het bovenstaande esdoornblad bijvoorbeeld, als een fractal kon simuleren. In zijn artikel 'Fractals made to order' en in zijn boek *Fractals Everywhere* besteedt hij er veel aandacht aan, waarbij hij ook vrij diep ingaat op de achterliggende wiskundige theorie. Zijn methode komt kort samengevat op het volgende neer. Hij probeert de gegeven figuur uit verkleinde kopieën van zichzelf weer aan elkaar te plakken als een collage. Zo'n kopie is in beginsel een gelijkvormige verkleining van het origineel in een verschoven en gedraaide positie. Maar men mag de kopie opvatten als iets elastisch dat nog in een bepaalde richting opgerekt kan worden, als een lineaire vervorming, kortom wat we in hoofdstuk 3 een affiene transformatie hebben genoemd. Bij de collage behoeft men niet al te precies te zijn. De samenstellende delen mogen elkaar een beetje overlappen en er mag ook enige ruimte tussen bestaan. Dat die methode succes heeft, wordt door Barnsley wiskundig bewezen en verantwoord in zijn zogenaamde 'collage theorema'.
Het bewuste esdoornblad kan bijvoorbeeld opgebouwd worden uit vier verkleiningen zoals in figuur 6.5.
Om de techniek in de vingers te krijgen, kan men het beste zelf eens een fractal met een min of meer voorgeschreven vorm ontwerpen.
We stellen ons tot taak een fractal te ontwerpen die de vorm heeft van een hoofdletter H. Om het rekenwerk zo eenvoudig mogelijk te houden, geven we die letter de vorm van een vierkant waaruit twee kleinere vierkanten zijn weggenomen, als getekend in figuur 6.6. De letter heeft dus de vorm van zeven identieke vierkanten.



Figuur 6.6 Een hoofdletter H



Figuur 6.7 Een typografische fractal

Het ligt voor de hand zeven afbeeldingen te nemen die elk het grote vierkant afbeelden op een van die deelvierkanten. Daartoe kunnen we volstaan met telkens een centrale vermenigvuldiging met de lineaire verkleiningsfactor 1/3. Met behulp van het programma COLLAGEH, weer een variant van het esdoornprogramma, komen we dan tot de fraaie, in figuur 6.7 afgebeelde, fractal. De in hoofdstuk 2 gegeven theorie stelt ons tevens in staat de fractaldimensie daarvan te berekenen als $\log(7)/\log(3) = 1.77$. Er zijn natuurlijk nog talloze andere manieren om de letter met kopieën van zichzelf op te vullen, maar we laten het aan de lezer over verdere experimenten uit te voeren.

Fractalspionage

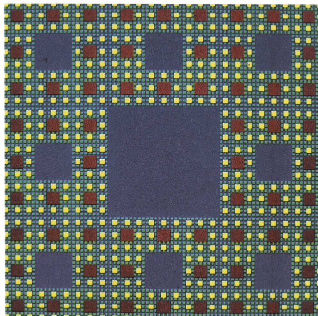
Wat we met een varenblad of met een letter kunnen doen, kunnen we met elk willekeurig meetkundig object uitvoeren. In gedachten zien we de elementen van een schilderij of van een foto al vervangen door een serie fractals. Men kan zich afvragen wat voor zin zoiets heeft. Maar laten we ons nu eens niet door artistieke overwegingen leiden en ons alleen bezighouden met documentatie. De gebruikelijke manier om de gegevens van een foto in een archief op te bergen, ter wille van latere raadpleging, berust op het aanbrengen van een denkbeeldig raster. De foto wordt bijvoorbeeld opgedeeld in een vierkant van 1000×1000 pixels en van elke pixel moet de grijswaarde of kleurwaarde bewaard worden. Het is bekend, of we kunnen het alleen maar vermoeden, dat er vele kunstmatige satellieten zijn die zich in het wereldruim bewegen met als enige opdracht het verzamelen van fotografisch materiaal. Van de hoeveelheid gegevens kan men zich nauwelijks een voorstelling maken, maar het is duidelijk dat het om gigantische hoeveelheden 'megabytes' gaat. Elke techniek die de voortdurende

stroom van gegevens wat kan indammen is welkom, en nu komen de fractals ons te hulp.
Stel dat het varenblad van Barnsley van figuur 3.4 onderdeel zou uitmaken van een foto. Voor de documentatie van deze plant zouden we dan alle samenstellende varenpixels in ons databestand moeten opnemen, een heel werk gezien het grote aantal details. We veronderstellen dat we voor de documentatie een lijst van zowat tienduizend getallen nodig hebben, waarbij een getal de betekenis heeft van een geheel getal tussen 0 en 99. Misschien is die taxatie nog aan de lage kant, maar dat hindert niet. Laten we nu eens kijken naar de fractaltegenhanger. De in het programma FERN2 bepaalde fractal behoeft voor zijn definitie niet meer dan de boven gegeven tabel van 4×7 coëfficiënten, het overige is het standaardprogramma. Het komt er dus eenvoudig op neer dat het aantal van tienduizend teruggebracht is tot slechts 28, een reductie met een factor van zowat 400.
Dit eenvoudige voorbeeld geeft een idee van wat voor mogelijkheden er zijn. Er is reeds een bedrijf gesticht waarin de ideeën van Barnsley en zijn medewerkers op commerciële wijze geëxploiteerd worden. Helaas heeft, zoals te verwachten was, ook de geheime dienst belangstelling gekregen, en aldus onttrekt veel research zich aan de openbaarheid. In grote lijnen is het niet eens zo moeilijk om een gegeven foto, een landschap of wat dan ook als een samenstel van fractals uiteen te rafelen. Laten we aannemen dat het gaat om een zwart-wit foto met alleen grijswaarden. Zijn er bijvoorbeeld zestien grijswaarden, dan kunnen we de afbeelding splitsen in een reeks van zestien stuks. Een individuele afbeelding bestaat uit een aantal vlekken, en die benaderen we zo goed mogelijk door ellipsen. Omdat twee ellipsen altijd door een affiene transformatie in elkaar overgevoerd kunnen worden, zijn het geschikte objecten om verder mee te werken. Het zoeken is nu vooral naar regelmatige series waarbij het herhalingsprincipe uitgebuit kan worden. Hoe meer herhalende elementen er zijn, des te beter het gaat. Men kan zich voorstellen dat bijvoorbeeld een veld met zonnebloemen zich uitstekend voor een dergelijke behandeling leent. Barnsley heeft dan ook verbluffende staaltjes van de door hem ontwikkelde techniek laten zien.

Een toekomstdroom

...voor de eerste keer was de mensheid erin geslaagd een onbemand ruimtevaartuig naar het dichtstbijzijnde planetenstelsel te sturen. Het ruimteschip was 'Mandelbrot' gedoopt, naar de twintigste-eeuwse pionier die aan de wieg had gestaan van de fractale verkenning. Met de nieuwe optische fractalisator zouden fraaie beelden van nog onbekende werelden verkregen kunnen worden, meer en beter dan vroeger met de klassieke methoden mogelijk was. De techniek van de fractale verkenning, tot dusverre voorbehouden aan de militairen, was nu voor het eerst voor astronomische doeleinden beschikbaar gekomen. Grote opwinding ontstond er toen de ontvangen beelden van een nog onbekende planeet, Julia gedoopt, geanalyseerd werden. Men meende er, zij het enigszins ruw, de contouren van het bekende appelmannetje, de Mandelbrot set, in te herkennen. Was dit een mogelijk eerste contact met ander intelligent leven in het heelal, dat

zich aldus aan de buitenwereld kenbaar wilde maken, of was het een toevallig verschijnsel zoals de kanalen van Mars? Men herinnerde zich dat eeuwen geleden al geopperd was om de stelling van Pythagoras in de Siberische wouden aan te brengen op een dusdanig reusachtige schaal, dat buitenaardse bezoekers dit wel moesten opmerken als een uiting van intelligentie. In de zeventiende druk van de Grote Winkler Prins staat bij het trefwoord 'practical joke' de volgende passage: '...een bekend voorbeeld van een practical joke is de afbeelding van de Mandelbrot set welke een computerprogrammeur verwerkte in de software van een astronomisch verkenningsprogramma, hetgeen indertijd tot grote opwinding leidde...'



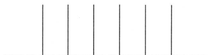
De zeef van Menger

Stochastische fractals

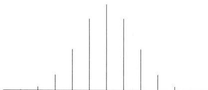
In de natuur komen we een grote verscheidenheid aan grillige meetkundige patronen tegen waaraan we de betekenis van een fractal kunnen geven. We stuiten daarbij op een interessant filosofisch probleem. Een denneboom is voor ons een vertrouwd begrip, en menig exemplaar, puur natuur of kunst, bevindt zich aan het eind van het jaar in onze woning. Geen twee exemplaren zijn exact gelijk, maar toch hebben we geen enkele moeite een spar of den van andere boomsoorten te onderscheiden. Blijkbaar bevindt zich in onze geest een beeld van een ideale den, een soort wiskundige structuur waaraan de ideale boom behoort te voldoen. Het meest opvallende in dergelijke structuren is het element van zelfgelijkvormigheid, geen zelfgelijkvormigheid volgens een strak wiskundig schema, maar meer in de zin van toevallige overeenkomsten en gelijkenissen in verschillende schalen.
Liever dan het houden van abstracte redeneringen geven we een paar concrete voorbeelden. Een kustlijn ziet er op grote schaal grofweg net zo grillig uit als op kleine schaal. De grote golven van de oceaan maken op een zich in een vliegtuig bevindende beschouwer dezelfde indruk als de waterrimpels op een hengelaar in een vissersbootje. De door Robert Brown waargenomen toevallige beweging van stuifmeelkorrels in een waterdruppel onder de microscoop is het prototype van een zogenaamde stochastische fractal. Het is een fractal, omdat de structuur van de waargenomen beweging niet verandert wanneer de schaal verandert. Het ziet er eender uit ongeacht welke vergroting Brown zou gebruiken, en dat is dus zelfgelijkvormigheid, de essentie van het begrip fractal. Het adjectief 'stochastisch' wil zeggen dat de waargenomen structuur onderworpen is aan de wetten van het toeval, een bewering waaraan overigens veel vastzit, maar ook daaraan gaan we voorbij, omdat een nauwkeurige beschrijving een grondige kennis van de waarschijnlijkheidstheorie vereist. Het is overigens interessant te weten dat juist Mandelbrot vanuit die theorie, de zogenaamde stochastiek, het fractalbegrip wiskundig heeft opgebouwd.

Zuiver gokken

Van de waarschijnlijkheidsrekening hebben we niet zoveel technische kennis nodig om er enigszins mee te kunnen werken. Van belang is vooral het begrip van een verdelingsfunctie. Een zuivere dobbelsteen kan zes uitkomsten geven: 1, 2, 3, 4, 5, 6, die alle even waarschijnlijk zijn. Omdat er zes mogelijkheden zijn heeft elke worp de kans 1/6. Grafisch geven we dat



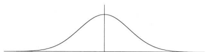
Figuur 7.1 De verdelingsfunctie van de worp van een dobbelsteen



Figuur 7.2 De verdelingsfunctie van tienmaal kruis of munt



Figuur 7.3 De verdelingsfunctie van RND



Figuur 7.4 De normaalverdeling

weer in figuur 7.1, waarin verticaal de kansfunctie is uitgezet. Die heeft voor elk der getallen 1, 2, 3, 4, 5 en 6 de waarde 1/6, en we spreken van een discrete kansverdelingsfunctie. De toevoeging discreet wil alleen maar zeggen dat er een beperkt aantal uitkomsten is. Het resultaat van het werpen van een dobbelsteen is een voorbeeld van een zogenaamde stochastische grootheid. Zo'n grootheid kan in dit geval dus zes waarden aannemen met gelijke kans.
Het werpen van een zuivere munt, kruis of munt, levert een voorbeeld van een stochastische grootheid met twee uitkomsten, op te vatten als 0 of 1 met gelijke kans. Werpen we tienmaal achter elkaar met een zuivere munt, dan is het totaal van de worpen met kruis een voorbeeld van een stochastische variabele met elf uitkomsten, van 0 tot en met 10. De verdelingsfunctie van de mogelijke totalen ziet er nu uit als in figuur 7.2.
Naast een discrete stochastische grootheid staat een continue stochastische variabele. Een bekend voorbeeld is de RND-functie van computers, welke een willekeurig getal tussen 0 en 1 oplevert. In werkelijkheid gaat het om een discrete grootheid, omdat een computer nu eenmaal beperkingen heeft bij het voorstellen van getallen. Met de veel gebruikte precisie van zes decimalen is er tussen 0 en 1 slechts ruimte voor een miljoen computergetallen uit een oneindige hoeveelheid van mogelijkheden. Afgezien van deze beperkingen is de RND-functie een heel bruikbaar surrogaat voor een ideale continue stochastische grootheid. De bijbehorende verdelingsfunctie is grafisch niets anders dan een horizontaal lijntje zoals in figuur 7.3.

De normaalverdeling

De voor de praktijk belangrijkste verdelingsfunctie is de zogenaamde *normaalverdeling*. De bijbehorende stochastische variabele, die we x noemen, kan theoretisch alle waarden aannemen. De kans op een uitkomst van x welke tussen de grenzen x_1 en x_2 inligt, kunnen we grafisch uit de figuur aflezen als de grootte van het oppervlak dat begrensd wordt door de getallenlijn, de verticalen bij de grenzen en de verdelingsfunctie.
De normaalverdeling is gekenmerkt door een gemiddelde en een standaardafwijking of *strooiing*, traditioneel aangeduid met σ. Het kwadraat van de strooiing staat bekend als de *variantie*. De strooiing is een maat voor de breedte van de normaalverdeling. We nemen aan dat de stochasti-

sche variabele x voldoet aan de in figuur 7.4 afgebeelde normaalverdeling waarvoor het gemiddelde 0 is. De kans op een waarde van x binnen het interval $(-3\sigma,3\sigma)$ is gelijk aan 0.9973. Dat betekent dat een afwijking ten opzichte van het gemiddelde van meer dan driemaal de strooiing een grote zeldzaamheid is, minder dan drie promille. In de praktijk komt de normaalverdeling erg veel voor, en een groot deel van de klassieke statistiek is ook hierop gebouwd.
Bewezen kan worden dat de samenwerking van een groot aantal toevalligheden, het vele malen werpen van een munt, op den duur tot een normaalverdeling leidt, een wetmatigheid waar we bij computersimulatie ons voordeel mee kunnen doen. We kunnen dat al waarnemen bij de figuren 7.2 en 7.4, die inderdaad grote gelijkenis vertonen.

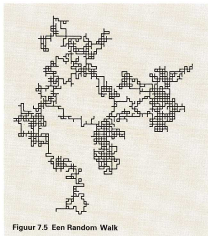
Brownse beweging

In hoofdstuk 1 hebben we al vermeld dat Robert Brown (1818) in zijn microscoop grillige bewegingen van stuifmeelkorrels waarnam, en concludeerde dat die beweging niet van biologische aard was, maar door fysische wetten bepaald. We kunnen ons voorstellen dat een stuifmeelkorrel ten gevolge van de warmtebeweging van de omringende moleculen telkens kleine tikjes uit willekeurige richtingen krijgt. De tikjes zijn klein en de korrel is relatief gesproken geweldig groot. Er is dan ook heel wat nodig om die logge massa uit balans te krijgen, maar toch gebeurt dat telkens, en het resultaat nemen we met een menselijke maat van tijd en ruimte waar als een onregelmatig trillende beweging. Een eeuw later heeft ook Einstein (1905) zich met de theorie van de Brownse beweging beziggehouden, en onder andere aangetoond dat de grootte van het deeltje er niet toe doet.
De baan van een puntvormig deeltje dat in een vlak een Brownse beweging uitvoert kunnen we nu opvatten als een lijnfractal die in statistische zin zelfgelijkvormig is op alle schalen. Dat betekent dat een statistische analyse van elk willekeurig fragment tot dezelfde statistische kenmerken aanleiding geeft. Het voordeel van deze interpretatie is dat voor een Brownse beweging de theorie van de fractaldimensie toegepast kan worden.

Random walk of de weg van een dronkeman

Men stelle zich een havenwijk voor die als een soort schaakbord bestaat uit vele lange, elkaar loodrecht kruisende, straten. Op de kruispunten bevinden zich kroegen waar het goed drinken is. Een dorstige zeeman die behoefte heeft aan wat afwisseling begeeft zich op weg, loopt een stukje in een willekeurige richting, bezoekt de dichtstbijzijnde gelegenheid, drinkt een glas en begeeft zich wederom op weg naar de volgende aanlegplaats. Dat gaat zo, althans in dit verhaal, eindeloos door. Telkens legt hij van kroeg tot kroeg in een willekeurige richting een willekeurige afstand af. Wat we beschreven hebben staat in de wandeling bekend als een 'dronkemansloop'. In de waarschijnlijkheidsrekening heet het wat deftiger een *random walk*, en is het een belangrijk deelgebied waar veel over gepubli-

ceerd is. Een door een punt op ruitjespapier beschreven random walk geeft een goede praktische benadering van een Brownse beweging. Ideaal zou zijn dat de ruitjes oneindig klein waren, maar helaas hebben we geen oneindig lange tijd tot onze beschikking.
Met het computerprogramma RWALK hebben we er een simulatie van gemaakt waarbij een punt als een dronkeman wordt losgelaten in het centrum van een rechthoekig lijnenrooster. Het aantal verticale en horizontale lijnen kan willekeurig gekozen worden, afhankelijk van het aantal pixels dat op het beeldscherm ter beschikking staat. Een aantal van 401 bij 301 is heel geschikt.
In het programma wordt door het rondzwervende punt telkens een stapje met een vaste grootte, bijvoorbeeld een roosterafstand, afgelegd. Op elk moment wordt er dus een keuze uit vier mogelijkheden gedaan, links, rechts, onder of boven. De keuze wordt telkens gemaakt met de RND-functie van de computer. Zoals men wellicht weet hangen de door de computer te genereren toevallige getallen af van een aan het begin van het programma zelf te kiezen begingetal (de 'seed' in computerjargon). Een zelfde begingetal geeft steeds dezelfde rij van 'toevallige' getallen.


Figuur 7.5 Een Random Walk

Echt toevallig zijn ze daarom natuurlijk niet, maar wel zijn ze daarvoor een goed surrogaat omdat ze zich ten aanzien van de gebruikelijke statistische toetsen in alle opzichten als toevallige getallen gedragen. Een illustratie als gegeven in figuur 7.5 hangt daarmee sterk af van dat gekozen begin getal. De lezer en computergebruiker wordt daarom sterk aangeraden met het bewuste programma vele malen te experimenteren en daarbij telkens een ander begin getal te kiezen. Desgewenst kan men ook de opdracht RANDOMIZE TIMER gebruiken, omdat daarbij het begin getal automatisch gekozen wordt, afhankelijk van de interne klok van de computer. In het programma wordt de baan in een van tevoren gekozen aantal stappen vervolgd. Maar wanneer de baan voor die tijd de rand van de rechthoekige begrenzing bereikt, stopt het programma automatisch.

In de theorie van de random walk spreekt men van absorberende wanden. In de toepassingen staat het verschijnsel model voor een materieel deeltje, een molecuul bijvoorbeeld, dat al rondzwervend in een medium zich ten slotte, door chemische of fysische binding, ergens aan hecht. Met een kleine wijziging in het programma kan men ervoor zorgen dat het deeltje niet aan de rand geabsorbeerd wordt, maar binnen de gegeven rechthoek blijft doorlopen. Men spreekt dan van spiegelende wanden. Het is alsof een experiment in een glazen bakje wordt uitgevoerd.
Het beschreven computerexperiment is het eerste en eenvoudigste van een

hele reeks soortgelijke modellen. Het kan op allerlei manieren verbeterd en verfijnd worden. Maar het valt daarbij wel op dat globaal gesproken het resultaat kwalitatief steeds hetzelfde is, zeker wanneer er vele kleine stapjes genomen worden. We hebben er bij het bespreken van de normaalverdeling al op gewezen dat het vele malen kruis of munt gooien al bij benadering tot de normaalverdeling leidt. Dat zelfde is het geval in het computerexperiment. Feitelijk wordt er bij elke stap tweemaal kruis of munt gegooid: 'links of rechts' en 'omhoog of omlaag'. Na een groot aantal van dergelijke stappen is het alsof het bewegende punt in een willekeurige richting een volgens de normaalverdeling bepaalde willekeurige stap gedaan heeft. Ook dit zouden we met hetzelfde computerexperiment kunnen navolgen. Daartoe zouden we de staplengte wat moeten verkleinen, zouden we bij de afbeelding telkens een aantal stappen kunnen overslaan, en zouden we het totale aantal stappen fors moeten verhogen. Voor een zeer snelle computer is dit nog wel een beetje te doen, maar toch worden ook daar de fysische beperkingen ten aanzien van ruimte en tijd snel bereikt. Het is daarom een gelukkige omstandigheid dat er juist op theoretisch gebied vele en belangrijke resultaten zijn behaald.

Diffusie

De door Brown ontdekte toevallige beweging van een stuifmeelkorrel is dus niets anders dan een random walk waarbij de individuele stapjes van moleculaire afmetingen zijn. De waarschijnlijkheidstheorie leert ons dat de statistische eigenschappen van de baan niet van de schaal afhangen. Er is dus gelijkvormigheid op alle schalen, en daarmee is de baan een echte fractal. We zouden ons kunnen afvragen wat de fractaldimensie is: het antwoord is 2. Dat wil zeggen dat de baan, statistisch gesproken, vlakvullend is. Computerexperimenten op de wijze van figuur 7.5 laten dat ook zien. Wanneer we maar lang genoeg wachten, komt de baan overal waar we maar willen. Een enkele baan ziet er overigens grillig genoeg uit, maar wanneer we een groot aantal deeltjes vanuit een zelfde beginpunt een random walk zouden laten beginnen, en wanneer we de deeltjes als totaliteit beschouwen, ziet het er veel regelmatiger uit. In de natuurkunde spreken we dan van *diffusie*. Dat verschijnsel kan betrekking hebben op moleculen of bijvoorbeeld kleine partikeltjes kleurstof. Wanneer we een druppel inkt laten vallen op een vel filtreerpapier of vloeipapier, verspreiden de individuele kleurstofdeeltjes zich langs de poriën van het papier in een random walk. Macroscopisch merken we er niet veel van. We zien alleen dat de kleurstof zich verspreidt in steeds grotere cirkels.
In de mathematische fysica wordt de diffusie van materie in een poreus medium bepaald door de differentiaalvergelijking

$$D\Delta u = \delta u / \delta t,$$

waarbij u de concentratie van de materie, t de tijd en D de zogenaamde diffusieconstante is. Het symbool Δ is de operator van Laplace. Over de betekenis van dat symbool en de afleiding van deze vergelijking raadplege men de wiskundige literatuur. Hier vermelden we alleen dat de diffusie

van materie langs een enkele oneindig lange lijn, de x-as, bepaald wordt door de zogenaamde puntbronoplossing

$$u = 1/(2\sqrt{\pi Dt}\exp(-x^2/(4Dt))$$

Die oplossing kunnen we grafisch interpreteren als een normaalverdeling met gemiddelde 0 en strooiing $\sqrt{2Dt}$. Op het begintijdstip is alle massa geconcentreerd in de oorsprong $x = 0$, daarna verspreidt de massa zich naar links en rechts. Op elk tijdstip is de massa langs de x-as verdeeld volgens de normaalverdeling. Interessant is de conclusie dat de strooiing, een maat voor de breedte van de massaverdeling, toeneemt met de wortel van de tijd.

In twee en in drie dimensies hebben we iets soortgelijks. Een hoeveelheid materie die op een begintijdstip geconcentreerd is in een punt van een vlak verspreidt zich in concentrische cirkels, waarbij de mate van uitbreiding weer bepaald is door dezelfde formule $\sqrt{2Dt}$.

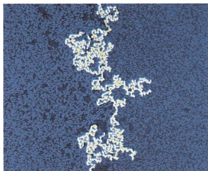
Ten gerieve van lezers die moeite hebben met het 'lezen' van wiskundige formules vatten we een paar essentiële punten samen. Diffusie is een heel algemeen verschijnsel, dat zich in onze wereld overal manifesteert waar sprake is van een ongerichte, chaotische of niet-systematische beweging van kleine deeltjes. Dat kunnen moleculen zijn, radioactieve elementen, giftige bestanddelen als dioxine, water en olie in poreuze aardlagen enzovoort. De deeltjes kunnen zich bewegen in water, in lucht, maar ook ondergronds via grondwater of doorlatende aardlagen. Uit deze opsomming zal wel duidelijk zijn dat er ter wille van een goed beheer van ons leefmilieu heel wat problemen op te lossen zijn. Langs natuurkundige en wiskundige weg kunnen we met behulp van zogenaamde wiskundige modellen beter inzicht krijgen. Het blijkt dat materie vanuit een beginconcentratie zich uitbreidt, waarbij de afstand toeneemt met de wortel uit de tijd. Diffusie kunnen we opvatten als een gelijktijdige random walk of Brownse beweging van vele deeltjes, gevolgd door een middelingsproces.

Er is een aantal uiterst belangrijke diffusieproblemen waaraan men tot voor kort niet zoveel kon doen. Maar dat is anders geworden nu de moderne computers een betere simulatie van natuurlijke processen mogelijk maken. En nu blijken als vanzelf ook de fractals een belangrijke theoretische ondersteuning te bieden. Op vele terreinen van industrie en wetenschap wordt er hard gewerkt. Nog veel verbergt zich in de laboratoria en de studeervertrekken. Maar gezien het grote belang proberen we met twee representatieve voorbeelden de lezer deelgenoot te maken van de moderne ontwikkelingen.

Natte vingerwerk

Het exploiteren van een olieveld is geen eenvoudige zaak. Men stelle zich voor dat ergens in de diepte een nog niet ontgonnen, cirkelvormig olieveld aanwezig is. De olie bevindt zich in de vorm van kleine druppeltjes in de poriën van een laag gesteente, zandsteen bijvoorbeeld. Van nature staat de olie daarin onder grote druk. Die druk, allereerst afkomstig van de hoger gelegen aardlagen, wordt doorgegeven aan de olie in samenwerking

met eveneens aanwezige hoeveelheden gas en water. Maar laten we aannemen dat gas afwezig is, zodat alleen water als drijvende kracht in aanmerking komt. Brengen we een boorgat aan in het midden van het veld, dan heeft de olie de neiging via allerlei microscopische kanaaltjes naar het boorgat toe te stromen en aan de oppervlakte te komen. Heeft de olie echter een nogal stroperige consistentie, dan gaat het niet zo vlot en moet er gepompt worden. Men zou denken, hoe harder er gepompt wordt des te meer olie komt eruit. Maar dat is in het algemeen niet waar. Het blijkt dat er steeds meer water bijkomt naarmate er harder gepompt wordt. De verklaring is dat waterdeeltjes zich veel gemakkelijker door de kanaaltjes van het poreuze gesteente heen bewegen dan taaie oliedruppels, dat de laatste wellicht opzij gedrongen worden in de race naar het boorgat. Experimenteel onderzoek in het begin van de vijftiger jaren van kleine laboratoriummodellen wees uit dat in een poreuze laag de grenslijn tussen water en olie helemaal niet zo regelmatig is, maar dat het water zich 'met lange, dunne vingers' in het oliegebied indringt. Met dit verschijnsel voor ogen kunnen we ons nu een betere voorstelling maken van wat er onder de grond gebeurt wanneer we te hard pompen. Kennelijk trekken we lange watervingers aan, de olie blijft nagenoeg zitten en eigenlijk pompen we alleen wat 'vet' water.
In de computerindustrie heeft men getracht op verschillende manieren dat ongewenste verschijnsel te ondervangen, teneinde toch zo goedkoop mogelijk zoveel mogelijk olie uit een dergelijk veld te kunnen halen. Dat hierbij theorie gesteund door computersimulaties kan helpen spreekt welhaast vanzelf. In zijn boek *Fractals* gaat Jens Feder, een natuurkundige van de Universiteit van Oslo, daar uitvoerig op in, en beschrijft hij onderzoek dat hij in samenwerking met de Noorse olie-industrie (Statoil) heeft verricht. Hij laat onder andere zien dat de grens tussen olie en water, die tegelijk in een vlakke, poreuze laag aanwezig zijn, een fractallijn is en dat de fractaldimensie daarvan ongeveer 1.75 bedraagt. In figuur 7.6 tonen we het resultaat van een computerexperiment dat mèn op een snelle personal computer zelf zou kunnen uitvoeren. Het uitgangspunt is een rechthoekige matrix van 400×300 pixels. Aan de linkerrand worden waterdeeltjes geïnjecteerd die daarna in de rechthoek door diffusie hun weg zoeken. In de praktijk imiteren we dat proces door tachtigduizend deeltjes op toevallige wijze over de rechthoek te verdelen. Horizontaal nemen we een normaalverdeling met het gemiddelde op de linkerrand en met een zodanige spreiding, dat slechts enkele deeltjes zich helemaal rechts bevinden. Alle overblijvende pixels vatten we op als minuscule oliedeeltjes. Het gaat nu om de bepaling van de grens tussen water en olie. In de fysische werkelijkheid moeten we rekening houden met geïsoleerde waterdeeltjes die gevangen zijn in de olie, of andersom. Om dergelijke eilandjes uit te schakelen, moeten we van de verzameling waterdruppels het samenhangende deel bepalen, een programmatechnisch gesproken lastig probleempje. Evenzo moeten we het samenhangende gedeelte van de olie bepalen. Daarna is het makkelijk, alle pixels van het samenhangende watergebied welke grenzen aan ten minste een pixel van het samenhangende oliegebied behoren tot de water-olie grens, de zogenaamde omhulling (hull) van de diffunderende watervloeistof.

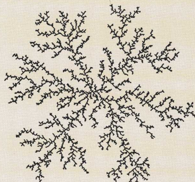


Figuur 7.6 De fractalgrens tussen olie en water in een computermodel

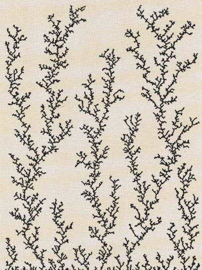
Het desbetreffende programma HULL is gezien zijn lengte en de specialistische betekenis niet in het boek opgenomen.

Fractalbliksem

Waarschijnlijk kent men wel de merkwaardige, stervormig vertakte patronen die het gevolg zijn van elektrische ontladingen, blikseminslag bijvoorbeeld. In figuur 7.7 zien we zoiets, maar de afgebeelde figuur is slechts het resultaat van een computersimulatie. Zulke stervormige dendrieten, toevallige fractalachtige vertakkingen die uitstralen vanuit een enkel middelpunt, of varianten ervan, komen in de natuur veel voor. Een ander voorbeeld vinden we in de kristallografie, waar op min of meer toevallige wijze halfgeordende samenklonteringen ontstaan, bijvoorbeeld van elektrolytisch gevormd zink. Dergelijke vormsels kunnen zowel in twee als in drie dimensies optreden. In drie dimensies kunnen dendrietachtige aggregaten ontstaan, wanneer bijvoorbeeld eiwitten zich in principe vrij bewegen, diffusie of random walk, maar aan elkaar vast gaan zitten bij een toevallige ontmoeting. Het verschijnsel staat in de vakliteratuur bekend als 'diffusion limited aggregation' of kortweg DLA.
Het programma CLUSTERC dat aan figuur 7.7 ten grondslag ligt, gaat uit van een door een cirkel begrensd pixelrooster. We beginnen met een deeltje in het centrum. Vervolgens wordt een tweede deeltje op toevallige wijze geplaatst in het cirkelvormige gebied, voldoende ver van het middelpunt, en dat deeltje laten we een Brownse beweging uitvoeren, een roosterstapje naar links, rechts, onder of boven met gelijke kansen. Met dit enigszins primitieve, maar voor de computer uiterst handige, proces imiteren we dus de diffusie. Komt het deeltje op de cirkelrand, dan beginnen we gewoon opnieuw, net zolang totdat het deeltje vlakbij het reeds in het middelpunt aanwezige deeltje komt. Het hecht zich daaraan vast, dat wil zeggen het blijft nu op zijn plaats, en we gaan verder met een volgend deeltje. Zo gaan we vele duizenden keren door. Telkens creëren we een deeltje en we laten dat net zolang diffunderen, totdat het zich vasthecht aan datgene wat reeds gevormd was. Het programma is betrekkelijk eenvoudig, maar zeer tijdrovend.



Figuur 7.7 Een elektrische ontlading van de computer



Figuur 7.8 Synthetische DLA-groeisels

Lange vingers

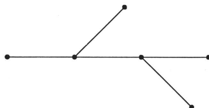
Bij ons verhaal over de produktie van olie en water hebben we erop gewezen dat een dunnere vloeistof zich met dunne, kronkelige vingers tussen de dikkere vloeistof kan gaan bewegen. Dat verschijnsel kunnen we ook begrijpen aan de hand van het zojuist beschreven computerexperiment. Alleen moeten we dan niet uitgaan van een punt in een centrale positie, maar van een rechte lijn. In figuur 7.8 hebben we een dergelijke situatie afgebeeld. Het diffusieproces vindt plaats in een staande rechthoek. Deeltjes diffunderen en blijven aan elkaar en ook aan de onderrand kleven. Het desbetreffende programma CLUSTERL is een eenvoudige variant van het vorige. Ook dit programma is eenvoudig en tegelijk tijdrovend. Wie dergelijke plaatjes ziet, denkt wellicht aan de chemische tuinen waarmee de scheikundedocenten hun gehoor plegen te vermaken. Anderen denken misschien aan de dendrietachtige figuren in mosagaat, een bekende halfedelsteen, of aan soortgelijke patronen van pyriet in zandsteen of kalksteen.

Lindenmayer-systemen

De Nederlandse bioloog A. Lindenmayer, tot aan zijn overlijden in 1989 hoogleraar te Utrecht in de theoretische biologie, heeft zijn naam gegeven aan een algemene wiskundige constructie om de structuur van planten te beschrijven. In dit hoofdstuk beperken we ons tot enkele eenvoudige voorbeelden van plantachtige patronen in twee dimensies.
Een onderdeel van een plant ziet er dan uit als een rij van symbolen

$...L [+ L] L [- L] L...$

De betekenis van de symbolen is als volgt:
L geeft een takje aan als een recht lijnstukje in een reeds bekende richting,
[geeft het begin van een zijtak aan,
$+$ bepaalt de richtingsverandering van een zijtak naar links,
$-$ bepaalt de richtingsverandering van een zijtak naar rechts,
] is het einde van een tak.



Figuur 8.1 Het basispatroon van een struik

Aan de bovenstaande symbolenrij kunnen we een grafische voorstelling verbinden door bij de $+$ en de $-$ bepaalde hoeken te kiezen, bijvoorbeeld 45 en -45 graden. Voor elke L tekenen we telkens een recht lijntje met een vaste grootte. Geven we de eerste L aan door een verticaal lijntje, als de stam van een boom, en plaatsen we de volgende lijntjes gewoon kop aan staart als bij het optellen van vectoren, dan ontstaat figuur 8.1.

De substitutieregel

De kern van de methode van Lindenmayer is het substitutieprincipe, een abstracte vorm van plantengroei. Een groeifase wordt in dit systeem beschreven door bijvoorbeeld alle letters L te vervangen door een vast patroon. We spreken daarbij van een *produktieregel*. Ter illustratie nemen we het volgende voorschrift:

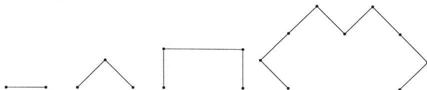
$L \rightarrow +L-L+$

Natuurlijk moeten we ergens mee beginnen, en dat doen we zo eenvoudig mogelijk, met een enkele letter *L*. Dus

fase 1	L
fase 2	+L–L+
fase 3	++L–L+–+L–L++
fase 4	+++L–L+–+L–L++–++L–L+–+L–L+++

Bij die substitutie blijven de symbolen + en – onveranderd. In dit heel eenvoudige voorbeeld doen de vertakkingssymbolen [en] niet mee. Laten we nu eens bekijken wat dit eenvoudige voorbeeldje meetkundig voorstelt. Daartoe interpreteren we *L* weer als 'teken een lijnstukje', en geven we + de betekenis van een richtingsverandering in positieve zin over 45 graden en laten we het symbool – corresponderen met een kwartslag naar rechts, de negatieve draaizin. De vier groeifases hierboven als 'woorden' beschreven, kunnen we nu meetkundig illustreren als in figuur 8.2.
We ontdekken, wellicht tot onze verrassing, dat we bezig zijn de opvolgende fases van een fractal te tekenen. In dit geval is het de welbekende lijn van Lévy. We begrijpen nu dat de substitutieregel in de 'taal' van Lindenmayer een abstracte en recursieve versie is van het principe van zelfgelijkvormigheid van fractals.
De systemen van Lindenmayer kennen veel mogelijkheden, maar in beginsel komt het altijd neer op het vormen van woorden met behulp van een beperkt aantal symbolen, het alfabet, en op het stapsgewijs uitvoeren van een produktieregel waarbij simultaan alle symbolen door iets anders vervangen worden. Dat kan hetzelfde symbool zijn, maar het kan ook een woord zijn.

Figuur 8.2 Groeifases van een kronkellijn



De kruipende schildpad

In de wereld van de computergrafiek is de term 'turtlegraphics' welbekend. Men stelt zich daarbij voor dat een kronkellijn getekend wordt door een robotschildpad die slechts een paar opdrachten kan uitvoeren. Die commando's zijn in beginsel 'vooruit en teken een lijn', 'linksaf' en 'rechtsaf'. Op ruitjespapier kan er aldus heel wat getekend worden. Velen van ons kennen ook een opgave als het natekenen van een gegeven figuur zonder de pen van het papier te halen. Soms wordt een dergelijk probleempje nog gecompliceerd door de aanvullende eis dat geen lijntje meer dan eenmaal getekend mag worden. Wanneer we bij de richtingsverandering van de schildpad willekeurige maar vaste hoeken, zoals bij het teke-

nen van de lijn van Lévy, toelaten, kunnen alle woorden van Lindenmayer zonder vertakkingssymbolen door de schildpad getekend worden. In het programma LEVYLIND is dit nader uitgewerkt. Dit programma, prototype van soortgelijke programma's, bestaat uit twee delen. In het eerste gedeelte wordt voor elke groeifase het nieuwe woord bepaald. Het oude woord bevindt zich in de array $x(.)$, terwijl het nieuwe woord in de array $y(.)$ opgebouwd wordt. In het bewuste programma zijn de elementen van deze arrays gehele getallen. De lengte van deze arrays kan enorm toenemen bij verdere groeifases, en de beperktheid van het werkgeheugen laat zich daarbij spoedig gevoelen. Uiteraard kan een handige programmeur daar wel wat op vinden, maar in het huidige programma kunnen we toch nauwelijks verder komen dan tien fases. In het tweede gedeelte van het programma worden de symbolen van het laatst gevormde woord symbool voor symbool geanalyseerd en grafisch verwerkt, een vrij eenvoudig proces.
Dat de lijn van Von Koch op een analoge manier gevormd kan worden, ligt nu voor de hand. Weer wordt een Lindenmayer-woord gevormd uit dezelfde symbolen, maar nu behoort bij + een hoek van 60 graden en bij – een hoek van – 120 graden. Het begin is weer een enkel lijnstuk L, en de produktieregel is (zie figuur 8.3 en het programma KOCHLIND)

$$L \rightarrow L + L - L + L$$

terwijl de hoeksymbolen geen verandering ondergaan.

De zeef van Sierpinski met de schildpad

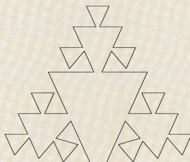
Dat de bekende Sierpinski-driehoek ook als een ononderbroken kronkellijn opgevat kan worden, ligt minder voor de hand. De structuur is dan ook iets ingewikkelder, en we moeten een extra symbool X erbij halen dat weliswaar geen directe grafische betekenis heeft, maar dat bepaalt waar iets gesubstitueerd moet worden. Het alfabet bestaat nu uit de vier symbolen L, X, + en –. Als beginwoord kiezen we weer zo eenvoudig mogelijk de enkele letter X, grafisch een punt. De produktieregel heeft nu betrekking op zowel L als X:

$$L \rightarrow L\ L$$
$$X \rightarrow -\ L\ X\ L + L\ X\ L + L\ X\ L\ -$$

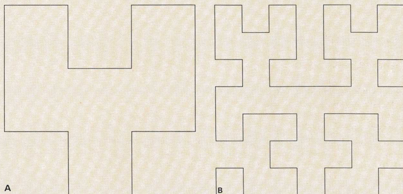
De symbolen + en – hebben nu de betekenis van een draaiing over 120 graden in positieve en in negatieve zin. Wat de produktieregel van X betekent kunnen we aflezen in figuur 8.4, waarin de tweede fase is afgebeeld. Ter wille van de duidelijkheid zijn daarin hoeken gebruikt die iets van 120 graden afwijken. Het resultaat is een stelsel gelijkzijdige driehoeken. De middens van de zijden zijn de X posities waar telkens gesubstitueerd wordt.
Het desbetreffende programma SIERLIND wijkt slechts weinig van de vorige programma's af. Het resultaat is natuurlijk bekend, maar het is aardig en instructief om op het beeldscherm te zien hoe de zeef als een ononderbroken kronkellijn gevormd wordt.



Figuur 8.3 Het schema van de Von Koch-lijn



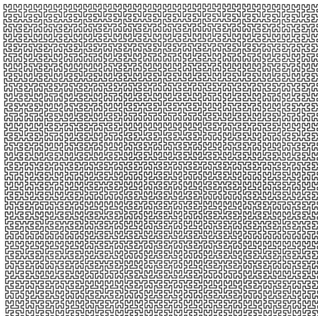
Figuur 8.4 De tweede groeifase van de Sierpinski-lijn



Figuur 8.5 Twee vroege stadia van een Peano-lijn

Een lijn van Peano

De Italiaanse wiskundige Giuseppe Peano (1858-1932) hield zich vooral bezig met de verzamelingenleer, een onderwerp nauw verbonden met de grondslagen van de wiskunde. Men zou hem nu misschien een voorloper van Mandelbrot kunnen noemen. De verdienste van Mandelbrot is vooral dat hij velen bewust heeft gemaakt van het bestaan van ingewikkelde meetkundige figuren, nu fractals genoemd. Maar Peano wees er in 1887 al op dat krommen zo ingewikkeld konden zijn, dat ze praktisch vlakvullend

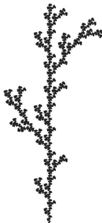


Figuur 8.6 De vijfde fase van een Peano-lijn

waren. Hij gaf er voorbeelden van en we spreken nog steeds van een Peano-kromme. In figuur 8.5 geven we twee vroege stadia in de vorming van een vierkantvullende Peano-kromme.
Het is geen erg gemakkelijk probleem om de systematiek in de ontstaanswijze van volgende fases te doorzien, en om die ook in een doelmatige produktieregel tot uitdrukking te brengen. We onthouden ons dan ook van een uitvoerige bespreking, en we vermelden zonder commentaar het substitutiemechanisme

$$X \rightarrow - Y L + X L X + L Y -$$
$$Y \rightarrow + X L - Y L Y - L X +$$

met de start X.
De hoekveranderingen zijn daarbij een kwartslag naar rechts, respectievelijk naar links. In het programma PEANOL is het een en ander uitgewerkt. Zonder extra voorzieningen kunnen we daarmee tot aan de vijfde groeifase komen. We krijgen dan de in figuur 8.5b getoonde kronkellijn. Om verder te komen, zijn er verschillende mogelijkheden. Heeft men de beschikking over een harde schijf, dan kan men de elementen van de x- en de y-array telkens naar een file op de harde schijf transporteren of er weer vanaf halen. Het werkgeheugen behoeft dan nauwelijks aangesproken te worden. Een op deze wijze vervaardigde illustratie is in figuur 8.6 afgebeeld.



Figuur 8.7 Een struikje in de vijfde groeifase

Vertakkingen

We keren weer terug tot het patroon van figuur 8.1. Gaan we uit van het element L en passen we telkens de volgende substitutieregel toe

$$L \rightarrow L [+ L] L [- L] L$$

dan krijgen we een snel groeiend woord dat in de vijfde fase al uit 6249 elementen bestaat. Interpreteren we de symbolen + en – als zijtakken die met de hoofdtak een hoek van ongeveer 28 graden (of bijvoorbeeld $\pi/7$ in radialen) maken, dan krijgen we het struikje van figuur 8.7. Het desbetreffende programma is BUSHLIND. In dat programma maken we, zoals we in hoofdstuk 5 reeds besproken hebben, gebruik van het principe van de stapel. Bij het interpreteren van het woord moeten we telkens bij het lezen van een symbool [de gegevens van het vertakkingspunt, positie van het knooppunt en richting van de vorige tak op een stapel leggen. Komen we bij het lezen een symbool] tegen, dan is het einde van een tak bereikt en moeten we terug naar de laatste vertakking. De gegevens daarvan bevinden zich bovenop de stapel en staan dus meteen tot onze beschikking. Ook hierbij doet zich het technische probleem voor dat de woorden van het Lindenmayer-systeem dusdanig snel aangroeien, dat al spoedig de computer de mededeling 'out of memory' geeft. Maar met enige ervaring in het programmeren is daar wel wat aan te doen, zij het ten koste van wat tijd.

Biologische relevantie

Biologen plegen hun materiaal op verschillende manieren te classificeren, en een van de methoden is om alleen op het uiterlijk te letten. Met de biologen stellen we ons hier op een morfologisch standpunt, waarbij we bij algen, zeewier, struikjes en bomen alleen op de vertakkingsstructuur letten. Er zijn dan twee hoofdtypen, de zogenaamde monopodie en de dichotomie. Bij de monopodie (letterlijk: uit een zelfde voet) is sprake van een hoofdas met zijassen. Dichotomie (in tweeën snijden) is in de botanie het verschijnsel dat een plantendeel zich in twee gelijke delen splitst, dus een vorkvertakking of een topsplitsing. Het is voor lagere planten als schimmels, wieren en varens de meest gebruikelijke wijze van groei. Bij hogere planten, zaadplanten dus, komt het veel minder voor en is monopodie de meest gebruikelijke vorm. Lindenmayer kon zijn methode goed toepassen op primitievere organismen, maar voor het beschrijven van structuren waar de monopodie overheerst, blijkt toch meer nodig te zijn. Bij dat laatste denken we vooral aan bomen als een ceder en een pijnboom, waar de hoofdstam als een mast van een schip recht omhoog rijst. In hoofdstuk 10, waarin we een aantal ruimtelijke fractals bespreken, komen we erop terug.

In 1984 beschreef Alvyn Ray Smith in het tijdschrift *Computer Graphics* een interessante uitbreiding van het boven beschreven systeem van Lindenmayer. Weer gaat het om het vormen van woorden opgebouwd uit een beperkt aantal symbolen. In zijn eenvoudigste vorm zijn het de symbolen 0, 1, [en], en daartoe beperken we ons hier. Het onderstaande voorbeeld is ontleend aan een artikel van Steve Estvanik (1986). De produktieregels hebben nu betrekking op tripels van nullen en enen. We kunnen uit de symbolen 0 en 1 de volgende acht drietallen vormen.
Het vormen van een volgende generatie betekent dat we aan elk van deze tripels een te substitueren patroon toekennen. Er zijn dus in totaal acht substitutieregels. Om de methodiek aan de hand van een concreet voorbeeld te kunnen toelichten, nemen we het volgende schema:

000→0	100→0
001→1	101→00[01]
010→0	110→0
011→1	111→0

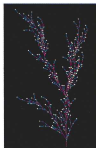
Een tripel wordt gevormd door bij een symbool zowel het vorige als het volgende symbool te betrekken. Is er geen vorig symbool aanwezig, dan nemen we daarvoor 1. Het volgende symbool moet zich wel aan dezelfde tak bevinden; zijn we reeds aan het einde van een tak, dan is het volgende symbool per definitie een 1. We zijn nu klaar om uitgaande van het symbool 1 telkens een volgende generatie, een nieuw woord, samen te stellen.

Generatie											
0	1										
tripel	111										
1	0										
tripel	101										
2	0	0	[	0	1	]					
tripel	100	001		001	011						
3	0	1	[	1	1	]					
tripel	101	011		111	111						
4	0	0	[	0	1	]	1	[	0	0	]
tripel	100	001		001	011		011		100	001	
5	0	1	[	1	1	]	1	[	0	1	]
tripel	101	011		111	111		111		101	011	

en ten slotte als de zesde generatie

0 0 [0 1] 1 [0 0] 0 [0 0 [01] 1]

Het ziet er allemaal nogal merkwaardig en gezocht uit, maar we hebben hierboven al gezien tot welke fraaie meetkundige fractalachtige patronen de woorden van een Lindenmayer-systeem konden leiden. De meetkundige interpretatie is hier overeenkomstig. De symbolen [en] duiden weer

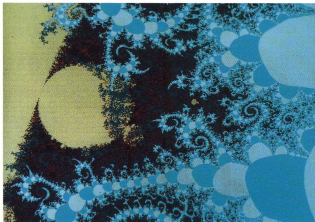


Figuur 8.8 Graftal-struikgewas

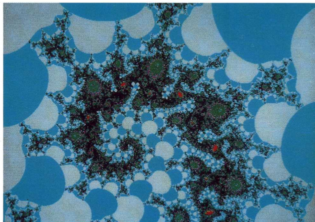
het begin of het einde van een zijtak aan. De hoeken die de zijtakken met de hoofdtak maken, moeten we nader vaststellen. Het eenvoudigste is het om alle hoeken gelijk te maken, of om die hoeken periodiek te kiezen uit zeg vier mogelijkheden. De symbolen 0 en 1 kunnen we lezen als het tekenen van een tak als een recht lijnstuk. De twee mogelijkheden 0 of 1 kunnen we gebruiken door de takken verschillend te kleuren. Een andere mogelijkheid is om ze een verschillende lengte te geven; kortom, er zijn heel wat interpretaties mogelijk.

In figuur 8.8 tonen we een volgens deze methode gevormde graftal, een soort struikgewas. Daarbij zijn we twaalf generaties diep gegaan. De arrays zijn dan al uitgedijd tot 2500 elementen.

Een sinus Mandelbrot set aangetast door een computervirus (vergrotingsfactor 15)



Spiraalstructuren in de sinus Mandelbrot set (vergrotingsfactor 150)



Cellulaire automaten

Hij werd in 1903 in Hongarije geboren als Janos, ging in 1926 naar Göttingen als Johann en emigreerde naar Amerika waar hij in 1937 als John staatsburger werd. We hebben het over Von Neumann, een der grootste wiskundigen uit onze tijd. Hij is in grote kring vooral bekend geworden als grondlegger van de theorie van de moderne rekenmachines. In zijn relatief korte leven, hij overleed in 1957, hield hij zich onder andere bezig met de problemen van het leven. Hij vroeg zich af of het mogelijk was een machine te bedenken die in staat was een exacte kopie van zichzelf te vervaardigen. Hij zag als het ware een robot, voorzien van een reeks instructies, zich door een pakhuis met onderdelen bewegen om daar materiaal te selecteren, en om dat later te combineren tot een nieuwe robot. Er zouden dan op een zeker moment twee robotten in werking zijn om een nieuwe kopie te maken, waarna de vier robotten dan... Het doet denken aan het gedicht 'Der Zauberlehrling' (1797) van Goethe, zo mooi uitgebeeld in de muziek van Paul Dukas als L'apprenti sorcier.

Een cellulaire automaat

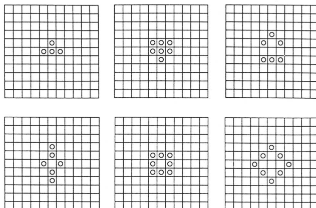
Als een goed wiskundige beschouwt Von Neumann voor zijn bewijs een zeer abstract wereldbeeld. Zijn 'wereld' bestaat uit een oneindig uitgestrekt vel ruitjespapier. Elk ruitje, door hem een *cel* genoemd, kan in een aantal zogenaamde toestanden verkeren. In de eenvoudigste versie zijn er slechts twee toestanden, die we als 'aan' of 'uit', als 1 of als 0, als actief of als passief kunnen opvatten. In zijn visie loopt de tijd met kleine stapjes, als het tikken van een klok. Op elke tik verandert het patroon van actieve en passieve cellen volgens bepaalde regels. Von Neumann kon langs deze weg met een aantal welgekozen rekenregels laten zien, dat het mogelijk was dat een zeker beginpatroon na een aantal tikken, dus aan zijn lot overgelaten en na verloop van tijd, zich als het ware had verdubbeld. Von Neumann werd bij zijn onderzoek geholpen door zijn trouwe vriend en collega Stanislaw Ulam, een Pools wiskundige, maar evenals Von Neumann het onzekere Europa ontvlucht. Voor zijn bewijs maakten zij gebruik van een model met 29 toestanden. Daarbij werd bij elke stap een cel slechts beïnvloed door zijn vier naaste buren.
Het werk van Von Neumann werd later door andere wetenschappers verder uitgewerkt, gevarieerd en vereenvoudigd. In 1970 wist de Engelse wiskundige John Conway er zelfs een soort spel, 'The game of life', van te maken. Zijn leefregels zijn de eenvoud zelve. Elke cel heeft acht buren,

vier aan de zijden en vier aan de hoekpunten. De spelregels zijn als volgt:

1 Een actieve cel omringd door één of geen actieve wordt passief.
2 Een actieve cel omringd door vier of meer actieve wordt passief.
3 Een passieve cel omringd door drie of meer actieve wordt actief.

In figuur 9.1 geven we een voorbeeld dat laat zien hoe een eenvoudig beginpatroontje zich in de loop van de tijd ontwikkelt.
Het door Conway bedachte spel heeft een groot succes gehad. Na de eerste beschrijving in de pagina's van de *Scientific American* (1970) door Martin Gardner, en de vele daarna volgende publikaties, heeft het velen in zijn ban gekregen. Het systeem leent zich bij uitstek voor computerbewerking, en menige bezitter van het moderne rekentuig is er reeds aan verslaafd geraakt. De grote charme van dit spel is de onvoorspelbaarheid. Een gedetailleerde wiskundige beschrijving is zo goed als onmogelijk, en het wetenschappelijk onderzoek is dan ook aangewezen op computerexperimenten. Aan Conway's spel kunnen we in dit bestek verder geen aandacht besteden, ook al omdat het weinig rechtstreeks verband heeft met fractals. Dat is wel het geval met een veel eenvoudiger model, dat onlangs in het tijdschrift *Algorithm* werd beschreven door Clifford A. Pickover.

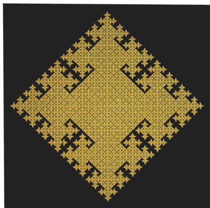
Figuur 9.1 Conway's game of life



Het groeimodel van Pickover

Uiteraard gaat dit model ook weer terug op de ideeën van Ulam en Von Neumann. Ook hier speelt het 'leven' zich af op een oneindig uitgestrekt vel ruitjespapier. Een cel kent slechts twee toestanden, 'levend' of 'dood', gesymboliseerd door 1 of 0. Een eenmaal levende cel heeft in dit model het eeuwige leven. Een dode cel komt tot leven wanneer in de vorige generatie precies een van de vier zijdelingse buren (links, rechts, onder, bo-

Figuur 9.2 De eerste generaties bij het model van Pickover



Figuur 9.3 Het patroon van Pickover na 126 generaties

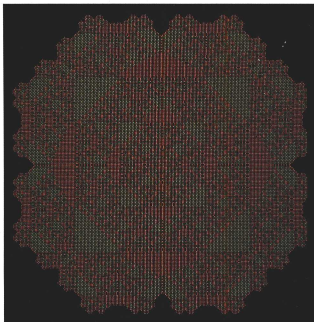
ven) leeft. Laten we eens beginnen met een enkele levende cel. In figuur 9.2 zijn een paar van de volgende generaties geschetst.
Het lijkt wel erg simpel, en zonder computer zouden we het al lang hebben laten rusten. Maar zie nu eens de situatie na 126 generaties, weergegeven in figuur 9.3. Dat plaatje is verkregen met behulp van het programma PICK1, een programma met een heel eenvoudige structuur. Om het beschikbare werkgeheugen van de computer optimaal te kunnen benutten, is bij dit programma dankbaar gebruik gemaakt van de aanwezige symmetrie. Slechts een kwart gedeelte van het patroon behoeft doorgerekend te worden. Het is een fantastisch gezicht te zien hoe op het beeldscherm het patroon zich uit een enkele levende cel ontwikkelt. Vierkanten vormen aan hun hoekpunten kleinere vierkanten op een wijze die karakteristiek is voor fractals.

Nog meer groei

Het besproken groeimodel van Pickover is het eerste van een hele reeks van soortgelijke modellen. In de eerste plaats kan men bij hetzelfde model onderzoeken wat er gebeurt wanneer we uitgaan van een klein groepje levende cellen. We laten het aan de lezer over daarmee te experimenteren. Met de eerste paar generaties lukt het nog wel met ruitjespapier, maar wil men meer en daarbij de computer te hulp roepen, dan is het voldoende het programma PICK1 een weinig aan te passen. Een andere methode is om de 'leefregel' te wijzigen. Bij het oorspronkelijke model kwam een cel tot leven zodra van de vier zijdelingse buren er precies één levend was. In plaats daarvan kunnen we ook de diagonale buren erbij betrekken. Een cel heeft nu acht buren, net zoals een koning midden op het schaakbord acht zetten kan doen. We kunnen dan als nieuwe 'leefregel' stellen, dat een cel tot leven komt zodra er van de acht buren precies één levend is. Pickover beschrijft een 'game of life' waarbij hij om en om van leefregel wisselt. Op een even tijdstip kiest hij de regel met de vier buren en op een oneven tijdstip die met acht buren. Zijn 'even-odd growth' is beschreven in het programma PICK2. Figuur 9.4 toont daarvan een kleurenversie, waarbij de kleur van een cel bepaald is door zijn aantal buren. Het desbetreffende programma, waaraan de lezer zelf gemakkelijk variaties kan aanbrengen, is PICK3.

Het is niet moeilijk zelf nog meer eenvoudige modellen te bedenken. De structuur van het programma is steeds dezelfde. Het patroon van een generatie is beschreven door een vierkante array $x(i,j)$, waarbij zowel i als j van 0 tot aan een zekere grens n, afhankelijk van het beschikbare computergeheugen, lopen. In feite is dit maar een kwadrant van het volle patroon dat zowel horizontaal als verticaal symmetrisch is. Voor de bepaling van de volgende generatie wordt voor elke dode cel i,j de omgeving onderzocht. We bepalen dus het aantal levende cellen van de vier zijdelingse buren of van alle acht buren. Is in het eenvoudigste geval dat aantal 1, dan komt in de volgende generatie de cel tot leven. De aldus bepaalde cellen worden opgeslagen in de array $y(i,j)$. Wanneer alle cellen aan de beurt zijn geweest, vindt generatiewisseling plaats en wordt $x(i,j)$ vervangen door $y(i,j)$.
We kunnen, net als Pickover deed, de twee genoemde leefregels om en om nemen, maar we kunnen natuurlijk ook een willekeurig ander ritme nemen, bijvoorbeeld tweemaal de ene en driemaal de andere enzovoort. Maar ook kunnen we bijvoorbeeld aan de diagonale buren een minder zwaar gewicht geven dan aan de zijdelingse buren. Weer een andere generalisatie is om aan de cellen meer toestanden toe te kennen, aan te geven met de getallen 0, 1, 2,... Op het beeldscherm zouden we elke toestand dan zijn eigen kleur kunnen geven. Genoeg mogelijkheden voor een computeraar om vele, vele uren leuk bezig te zijn.

Figuur 9.4 Een 'even-odd'-patroon in kleur



De driehoek van Pascal en die van Sierpinski

In 1883 beschreef Stephen Wolfram van de Universiteit van Illinois een nog eenvoudiger cellulaire automaat. De cellen strekken zich uit langs een rechte lijn en kunnen dus op eenvoudige wijze afgeteld worden als gehele getallen. Voorlopig nemen we aan dat er slechts twee toestanden zijn, dood (0) of levend (1). We beginnen met een eenvoudig voorbeeld. De 'leefregel' is als volgt:

Een cel omringd door een dode en een levende cel wordt levend.
Een cel omringd door of twee dode, of twee levende cellen wordt dood.

De toestand van de cel op de plaats i kunnen we aangeven als $x(i)$. De bovenstaande leefregel komt dan neer op

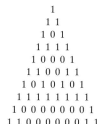
$$x(i) \rightarrow x(i-1) + x(i+1) \quad \text{mod } 2$$

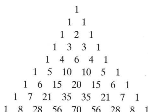
of uitgeschreven

$0+0=0$
$0+1=1$
$1+0=1$
$1+1=0$

Wie vertrouwd is met symbolische logica of Boole'se algebra herkent hierin een logische betrekking, een wetenschap die voor een computerprogramma handig kan zijn. In computertoepassingen werken we met een vast aantal cellen. We tellen dan van $i=0$ tot bijvoorbeeld $i=m$, wat neerkomt op $m+1$ cellen. Soms is het om redenen van symmetrie handig om te beginnen met een oneven aantal, en dan tellen we van 0 tot $2n$. De bovenstaande rekenregel behoeft nog een kleine aanvulling wanneer we aan het begin of aan het einde van de rij cellen zijn. Wanneer $i=0$, dan heeft de term $x(-1)$ natuurlijk geen zin, maar we doen dan net alsof die term 0 is. Iets overeenkomstigs doen we aan het eind van de rij en we rekenen dan $x(m+1)=0$. Wanneer we beginnen met een enkele levende cel en we plaatsen de opvolgende generaties onder elkaar, dan krijgen we een diagram als in figuur 9.5.
Wie vertrouwd is met de driehoek van Pascal herkent wellicht een zelfde structuur. Voor wie het niet meer zo precies weet geven we in figuur 9.6 een begingedeelte van die beroemde driehoek.

Figuur 9.5 De generaties van een enkele cel



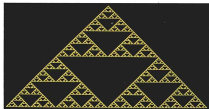


Figuur 9.6 De driehoek van Pascal

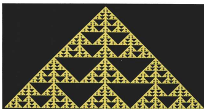
Zoals we zien is elk getal in die driehoek gelijk aan de som van de twee buren die er zich links en rechts boven bevinden. De getallen van de driehoek van Pascal zijn ook binomiaalcoëfficiënten, de coëfficiënten van de veelterm die ontstaat wanneer we $(a+b)$ tot een hogere macht verheffen. Wanneer we alle getallen van de driehoek van Pascal vervangen door hun rest bij deling door 2, anders gezegd wanneer we ze modulo 2 reduceren, dan krijgen we precies het cellulaire patroon van figuur 9.5.
De leefregel kan weer gemakkelijk in een computerprogramma worden verwerkt. In CELL1 is dat uitgevoerd. De toestanden van een generatie van $2n+1$ cellen is vastgelegd in de array $x(i)$ waarbij i van 0 tot $2n$ loopt. We beginnen met een rij waarin $x(n)=1$, terwijl de overige cellen in de toestand 0 zijn. De toestand van een cel van de volgende generatie wordt als $y(i)$ berekend met behulp van $x(i-1)$ en $x(i+1)$. Wanneer de nieuwe generatie helemaal gevormd is, wordt de oude array $x(i)$ vervangen door de zojuist gevormde $y(i)$ en beginnen we van voren af aan. Net als in figuur 8.5 worden de generaties op het beeldscherm rij voor rij onder elkaar geplaatst. Het is niet moeilijk om dan een paar honderd generaties door te rekenen. Het resultaat van een dergelijke berekening tonen we in figuur 9.7.
Wie enigszins vertrouwd is met fractals herkent hierin de bekende zeef van Sierpinski met zijn driehoekige gaten. En inderdaad, wanneer we het cellulaire patroon van figuur 9.7 oneindig lang zouden voortzetten en wanneer we tegelijkertijd de afmetingen zo schalen dat de cellendriehoek altijd dezelfde grootte heeft, dan is de limietfiguur identiek met de fractal van Sierpinski.
Kennelijk is hier iets heel bijzonders aan de hand. Wanneer we de driehoek van Sierpinski in de omgeving van een van zijn hoekpunten, de top bijvoorbeeld, voortdurend vergroten, verandert er maar heel weinig. De figuur is immers gelijkvormig met zichzelf en wel ten opzichte van elk hoekpunt met een schaling met factor 1/2. Bij elke vergroting met factor 2 is het totaal wel groter geworden, maar de omgeving van de top ziet er eender uit. We zouden kunnen zeggen dat door de verdubbeling een factor 1/2 ongedaan is gemaakt, maar er waren oneindig veel van die factoren, zodat we in feite niets opgeschoten zijn. Daarentegen laat de cellulaire automaat van figuur 9.7 ons die omgeving zien op een wijze die we vanuit de fractal werkend nooit te zien zouden kunnen krijgen. Anderzijds komen we na oneindigvoudige voortzetting van de automaat wel aan de complete fractal toe.
In figuur 9.8 laten we nog een interessante variant zien. De leefregel is nu

$$x(i) \rightarrow x(i-1)+x(i)+x(i+1) \quad \text{mod } 2$$

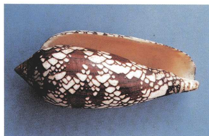
Het enige verschil is dat $x(i)$ nu ook meedoet. Drie opeenvolgende cellen doen nu elk in gelijke mate mee aan de eventuele vorming van een cel in de volgende generatie. Een dergelijke regel, waarin de totale som van de toestanden van alle in aanmerking komende cellen bepalend is, noemt Wolfram *totalistisch*.
Het desbetreffende programma CELL2 is maar een kleine variant van het vorige programma.



Figuur 9.7 Het resultaat van een enkele cel na 256 generaties



Figuur 9.8 Het resultaat wanneer we de regel iets veranderen

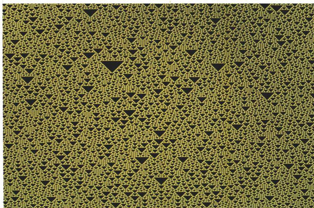


Figuur 9.9 Een *Conus textile* uit de Indische Oceaan

De levende en de levenloze natuur

In figuur 9.9 laten we een schelp zien, een bekende verzamelaarsschelp uit de kegelslakfamilie. De wetenschappelijke naam is *Conus textile*. Onze zeevarende voorouders moeten vele van dit soort fraaie schelpen op hun reizen in de Indische en de Stille Oceaan aangetroffen hebben. De bruinachtige tekening doet denken aan een Indische batikdoek en misschien is dat ook de reden voor de naamgeving.
Er zijn meer schelpen met een dergelijk patroon, maar deze is misschien de meest bekende.

We stellen ons nu een cellulaire automaat voor van het bovenbeschreven type, maar in plaats dat we uitgaan van een enkele cel als in figuur 9.7, beginnen we met een rij van cellen waarvan de aard door toeval bepaald is. Voor elke cel dobbelen we als het ware om te beslissen of hij actief of passief is. In het computerprogramma CELL1A is dat gedaan door met gebruikmaking van de RND-opdracht een toevallige rij van nullen en enen te genereren. Het resultaat is afgebeeld in figuur 9.10.
Het gevonden patroon doet sterk denken aan de decoratie van de *Conus textile*. Wanneer we nu ook nog weten dat het schelpenpatroon sterk variabel is, lijkt het welhaast of we achter een van de geheimen van de natuur gekomen zijn. We kunnen ons voorstellen dat een rij natuurlijke cellen in de biologische zin van het woord bezig is rij voor rij een patroon te weven door zich volgens een eenvoudige regel al of niet te dupliceren. Inderdaad is dit een van de vele voorbeelden dat ogenschijnlijk gecompliceerde zaken in de natuur heel goed het resultaat kunnen zijn van heel eenvoudige wetmatigheden.
Maar toch is voorzichtigheid geboden. Zodra we een eenvoudige wetmatigheid, in de vorm van een eenvoudig wiskundig model, gevonden hebben, zijn we nog lang niet klaar. Het is bepaald niet zo dat we daarmee de natuur verklaard hebben. Het kan zijn dat er meer modellen zijn die tot



Figuur 9.10 Een cellulaire automaat in actie

dezelfde resultaten leiden. En dat is stellig het geval bij de cellulaire automaat van figuur 9.10. Het blijkt namelijk dat er vele andere 'leefregels' zijn die tot een patroon leiden dat nauwelijks van dat van figuur 9.10 afwijkt.

De lineaire automaten van Wolfram

Met Wolfram beschouwen we weer een cellulaire automaat waarvan de cellen in twee toestanden, aangegeven door 0 of 1, langs een lijn gerangschikt zijn, een lineaire cellulaire automaat, zegt men. Het automatisme bestaat in een wijze van voortplanting waarbij telkens drie naburige cellen samenwerken om eventueel een nieuwe levende cel voort te brengen. Gebruiken we de bovenstaande notatie, waarbij $x(i)$ de toestand van een cel met rangnummer i aangeeft, dan zijn dus de toestanden van de drie cellen $x(i-1)$, $x(i)$, $x(i+1)$ bepalend voor de toestand van de cel $x(i)$ in de volgende generatie. Voor een groepje van drie cellen zijn er de volgende acht mogelijkheden

111 , 110 , 101 , 100 , 011 , 010 , 001 , 000

Een leefregel opstellen betekent dat we aan elk van die combinaties of een 0, of een 1 (een nieuwe cel in de volgende generatie) toekennen. Omdat er acht keuzemogelijkheden zijn, kunnen we 2^8 verschillende regels opstellen. Dat zijn er in totaal 256, maar niet alle mogelijkheden zijn even zinvol. Het is logisch om te veronderstellen dat de eerste combinatie 000 altijd tot een 0 moet leiden, en daarmee vervalt al de helft. Wolfram beschouwt van de overgebleven 128 mogelijkheden alleen die welke bij een symmetrische combinatie behoren, waarbij noch links noch rechts bevoorrecht wordt. Dat betekent in concreto dat een tweetal als 100 en 001 tot dezelfde toestand moeten leiden. Dergelijke regels noemt hij *legaal*, en men kan gemakkelijk nagaan dat er in totaal 32 van die regels opgesteld

kunnen worden. In de volgorde van het bovenstaande rijtje kunnen ze beschreven worden door de getallenrij

$a \quad b \quad c \quad d \quad b \quad e \quad d \quad 0$

In het computerprogramma CELL3 is dit rijtje samengevat in een enkel getal, de zogenaamde *code*. Dat getal is niets anders dan de interpretatie van het bovenstaande rijtje volgens het tweetallig talstelsel, dat wil zeggen

$$\text{code} = 128a + 64b + 32c + 16d + 8b + 4e + 2d$$

De eerder besproken eenvoudige regel, die tot de driehoek van Sierpinski leidt, correspondeert dus met de configuratie

0 1 0 1 1 0 1 0

of in code met het getal 90. Op analoge wijze correspondeert figuur 9.8 met de code 150.

Meer samenwerking

Het is niet moeilijk om nog algemenere cellulaire automaten te verzinnen, en Wolfram is ons al ver voorgegaan. Men kan meer cellen in de onderlinge samenwerking betrekken en men kan aan een cel meer toestanden toekennen. Wanneer telkens vijf cellen samenwerken, zijn er nog veel meer combinaties mogelijk, maar met Wolfram beperken we ons tot de zogenaamde totalistische regels. Dat zijn regels waarbij alleen het totaal van de waarden van vijf opeenvolgende cellen meetelt. Het gaat dus om de som

$$s = x(i-2) + x(i-1) + x(i) + x(i+1) + x(i+2)$$

Elke som draagt hiertoe op gelijke wijze bij. Zijn er alleen de toestanden 0 en 1, dan kan s de zes waarden 5, 4, 3, 2, 1, 0 aannemen. Omdat de waarde 0 alleen maar tot een volgende 0 kan leiden, kunnen we hiermee precies 32 verschillende regels samenstellen volgens het schema

5	4	3	2	1	0
a	b	c	d	e	0

De getallen a, b, c, d, e die elk 0 of 1 zijn, kunnen we weer onderbrengen in een code als

$$\text{code} = 32a + 16b + 8c + 4d + 2e$$

Zo correspondeert het patroon 101010 met het codegetal 42. In het programma CELLC kan men met een willekeurig zelf gekozen codegetal onderzoeken wat het resultaat is van een beginsituatie waarbij de toestanden toevallig gekozen zijn. Door telkens een andere code te kiezen krijgt men zo ongeveer alles tussen orde en chaos in met plaatjes die soms veel lijken op figuur 9.10.

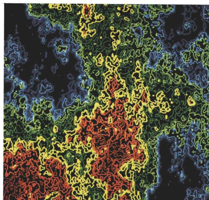
Wolfram heeft talloze mogelijkheden bestudeerd en zijn algemene conclusie is dat de evolutie van een cellulair systeem, dat gegenereerd wordt door een rij met een toevallig gekozen toestand, kwalitatief gesproken in vier klassen verdeeld kan worden.

1 Evolutie leidt tot een homogene situatie.
2 Evolutie leidt tot een aantal eenvoudige stabiele of periodieke structuren.
3 Evolutie leidt tot chaos met de structuur van een stochastische fractal.
4 Evolutie leidt tot een gecompliceerde structuur met een zekere permanentie.

Dit is overigens een nogal globale manier van indelen, maar we zien wel dat de ogenschijnlijk simpele, lineaire automaten van Wolfram veel weg hebben van dynamische systemen.
Een automaat van de eerste klasse is als een dynamisch systeem met een of meer stabiele evenwichten. De automaten met de codegetallen 0, 14, 16, 32, 36, 48, 54, 60, 62 behoren tot dat type.
Een automaat van de tweede klasse is als een dynamisch systeem met een aantrekkende periodieke cyclus. Voorbeelden zijn die met de codegetallen 8, 24, 40, 56, 58. De automaten van het derde en vierde type leveren limietstructuren die lijken op vreemde aantrekkers, op fractals met een gebroken dimensie. Tot die van het derde type rekent Wolfram de stelsels met de codes 6, 10, 12, 14, 18, 22, 26, 28, 30, 34, 38, 42, 44, 46, 50. Die met de codes 20 en 52 ten slotte zijn van het vierde type.

Ruimtelijke fractals

Fractals zijn niet gebonden aan een meetkundige dimensie. Er zijn al interessante fractals in één dimensie, op een rechte lijn bijvoorbeeld. Maar veel meer dan de bekende Cantor-verzameling is er toch nauwelijks te beleven. In twee dimensies, het platte vlak, is er een rijke wereld aan fractals, culminerend in de Julia-fractals en de bekende Mandelbrot set. Men kan dus verwachten dat er in de driedimensionale ruimte nog veel meer te beleven is, en dat is ontegenzeggelijk waar. Het probleem is evenwel dat we grote moeite hebben om van driedimensionale fractals een goede meetkundige voorstelling te maken. Computerschermen zijn nu eenmaal tweedimensionaal en kunnen ons slechts een projectie geven vanuit een of ander gezichtspunt. Het zou mooi zijn wanneer we als in een film om een ruimtefractal konden heendraaien, maar die techniek is slechts voorbehouden aan grote en snelle computers. Uiteraard is er veel mogelijk met de zogenaamde CAD-technieken (Computer Aided Design), en kunnen we met softwaregereedschap als Autocad een willekeurig ruimtelijk object als een bewegend voorwerp op het projectiescherm afbeelden. Ook kunnen we het object een willekeurige oppervlaktestructuur geven en het voorzien van een belichting in een willekeurige richting. Het is bijvoorbeeld niet moeilijk om een kubus als een soort maan te voorzien van een fractalachtig oppervlak met maankraters, en de hele zaak op kleurrijke wijze te belichten.



Een hoogtekaart van een fractallandschap

In dit hoofdstuk houden we ons liever bezig met meer aardse zaken en met wat we zelf, met een eenvoudige doch moderne computer, thuis tot stand kunnen brengen. In de wereld om ons heen valt overigens inspiratie genoeg op te doen. Sponzen en koralen doen ons aan fractals denken, met bomen hebben we ons al eerder beziggehouden, maar nu zijn we toe aan echte driedimensionale objecten uit de plantenwereld. Bloemkool, of liever broccoli, zouden we kunnen opvatten als eetbare fractals. Longweefsel, bloedvaten en zenuwdraden vertonen een duidelijke fractalstructuur. De hersenen zijn misschien wel de mooiste toepassing in de natuur van een ruimtelijke fractal. Men zou kunnen fantaseren dat de fractaldimensie van die structuur een objectieve maat is van intelligentie, veel beter dan een IQ verkregen met wat willekeurige tests.
In dit hoofdstuk beperken we ons tot enkele representatieve, eenvoudige voorbeelden. Alles wat we in dit hoofdstuk behandelen kan met betrekkelijk eenvoudige programma's op een computer nagevolgd worden. Maar in het algemeen geldt wel dat de verwerking van die programma's veel tijd kost. Gewenst is dat men beschikt over een beeldscherm met een goede resolutie, EGA (640×350) of VGA (640×480), en dat men zestien kleuren kan gebruiken. Speciaal bij het afbeelden van ruimtelijke fractals kunnen kleuren het voorstellingsvermogen krachtig ondersteunen.

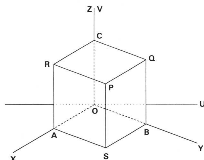
Projecteren

Bij het afbeelden van een ruimtelijke fractal moet men een meetkundige projectie uitvoeren, en daar is een beetje elementaire wiskunde voor nodig. We stellen onszelf de opgave een zich in de ruimte bevindende kubus op het beeldscherm af te beelden. Daartoe denken we de kubus vastgehecht aan een ruimtelijk x,y,z-coördinatenstelsel, gepositioneerd in de eerste ruimtehoek. De coördinaten van de acht hoekpunten zijn $O(0,0,0)$, $A(1,0,0)$, $B(0,1,0)$, $C(0,0,1)$, $P(1,1,1)$, $Q(0,1,1)$, $R(1,0,1)$, $S(1,1,0)$. In figuur 10.1 is afgebeeld wat we op het beeldscherm willen projecteren.
Die projectie is tot stand gekomen door de ruimtelijke x,y,z-coördinaten te vertalen in beeldschermcoördinaten die we met u,v aanduiden. De transformatieformules zien eruit als

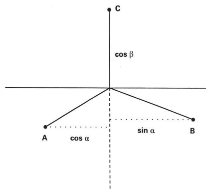
$$u = -x\cos(\alpha) + y\sin(\alpha)$$
$$v = -(x\sin(\alpha) + y\cos(\alpha))\sin(\beta) + z\cos(\beta)$$

waarbij α en β bepalend zijn voor de richting van de projectie. De betekenis van die hoeken blijkt wanneer we letten op de plaats van de hoekpunten O, A, B en C op het beeldscherm. Invullen van hun x,y,z-coördinaten in de projectieformules geeft:

$O(0,0,0)$ wordt geprojecteerd als $(0,0)$
$A(1,0,0)$ wordt geprojecteerd als $(-\cos(\alpha), -\sin(\alpha)\sin(\beta))$
$B(0,1,0)$ wordt geprojecteerd als $(\sin(\alpha), -\cos(\alpha)\sin(\beta))$
$C(0,0,1)$ wordt geprojecteerd als $(0,\cos(\beta))$



Figuur 10.1 De projectie van een kubus



Figuur 10.2 Van ruimte tot beeldscherm

Met behulp van figuur 10.2 zien we dan dat de hoek β de helling regelt van de *z*-as. Wanneer $\beta = 0$, dan loopt de *z*-as evenwijdig aan het projectievlak, het beeldscherm. Verticale afmetingen worden dan bij projectie op ware grootte afgebeeld. Wanneer β 90 graden is, dan staat de *z*-as loodrecht op het beeldscherm en wordt bij projectie als een enkel punt afgebeeld. De verticale afmetingen zijn dan geheel verloren gegaan. Bij een willekeurige hoek ertussenin treedt bij verticale lijnstukken een verkorting met de factor cos(β) op. De hoek α regelt de links-rechtsdraaiing. Houden we de hoek β vast en laten we α lopen van 0 tot +180 graden, dan krijgen we in de projectie de indruk dat de kubus om de *z*-as draait. We kunnen ook zeggen dat we de kubus als in rust beschouwen en dat we de kijkhoek, de richting waarin we de kubus bekijken, variëren. Het is dan alsof we om de kubus heenlopen.
In de praktijk maken we voor α vaak de speciale keuze van 45 graden. De punten *A* en *B* bevinden zich dan in een symmetrische positie links en rechts van de *z*-as. In dat geval zijn zowel cos(α) als sin(α) gelijk aan $1/\sqrt{2} = 0.7071$. Het coëfficiëntenschema van de transformatieformules kan dan vereenvoudigd worden tot

$$\begin{pmatrix} -1 & 1 & 0 \\ -\sin\beta & -\sin\beta & \sqrt{2}\cos\beta \end{pmatrix}$$

waarbij alle afmetingen met $\sqrt{2}$ vermenigvuldigd zijn.

Japanse bomen

Japanners hebben traditioneel een grote verering van de natuur. Hun tuinen en parken wekken alom bewondering en strekken tot voorbeeld in de westerse wereld. Japanners hebben er in hun cultuur steeds naar gestreefd de macrokosmos te belichamen in de microkosmos. Keurig aangeharkte

steentjes en een paar rotsblokken in een tempeltuin zijn voor hen een beeld van het eilandenrijk van Japan in de wereldzee. Een zelfde miniaturisatie treffen we aan bij de bekende Bonsai-boompjes, een kunstige schaalverkleining van wat in de natuur voorhanden is. We bespeuren al een begin van zelfgelijkvormigheid, het wezenskenmerk van de fractal. Dat de moderne Japanner zich met veel succes op de microelektronica heeft gestort, wordt daarmee begrijpelijker. De fractals hadden eigenlijk best in Japan uitgevonden kunnen zijn, maar een Japanse Mandelbrot was afwezig. Voorgangers en navolgers zijn er wel. Omstreeks 1900 ontwierp Teiji Takagi een wiskundige constructie, waarvan we nu zouden zeggen dat hij daarmee een der eerste fractals maakte. We komen er later in dit hoofdstuk nog op terug. Tegenwoordig staan in de Japanse wetenschappelijke wereld de fractals sterk in de belangstelling, vooral in verband met computer graphics.

In 1984 verscheen een artikel van Masaki Aono en Tosiyasu Kunii van de Universiteit van Tokyo getiteld 'Botanical tree image generation'. De auteurs beschrijven hoe op grond van een paar eenvoudige groeiwetten een computerprogramma opgesteld kan worden dat de uiterlijke vorm van een groot aantal boomtypen kan simuleren. Zij geven vele voorbeelden van gesimuleerde en van echte bomen, zoals de *Aucuba japonica* en de *Gingko biloba*.

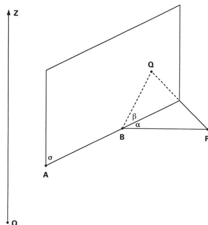
Wanneer we het artikel op meer wiskundige wijze bekijken, zien we dat we te maken hebben met een driedimensionale versie van de bij ons bekende boom van Pythagoras.

Een Japanse boom van Pythagoras

De groeiwetten kunnen aan de hand van figuur 10.3 als volgt worden beschreven. De hoofdas of stam van de boom is (verticaal) gericht volgens de richting van de zwaartekracht.

1 Een tak splitst zich in twee dochtertakken.
2 De lengte en de diameter van de dochtertakken staan tot die van de moedertak in een vaste verhouding.
3 Het vlak van twee dochtertakken staat loodrecht op het vlak gevormd door moedertak en de verticale richting. Het vlak van de dochtertakken van de stam staat verticaal.
4 Vertakking vindt simultaan plaats aan het eind van de takken.

We nemen aan dat AB een willekeurige tak is en dat die de hoek σ maakt met de verticale richting, de z-as OZ van het ruimtelijke coördinatenstelsel. In het punt B vertakt AB zich in de dochtertakken BP en BQ op zodanige wijze, dat A, B, P en Q in een zelfde vlak liggen. De vaste



Figuur 10.3 Het groeischema van de Japanse boom

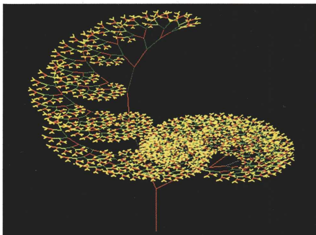
verkortingen noemen we p en q, zodat $BP = p \times AB$ en $BQ = q \times AB$. We nemen aan dat de hoeken die BP en BQ als vector beschouwd met AB maken, voor alle takken gelijk zijn aan respectievelijk α en β. Verder nemen we aan dat het vlak BPQ ten opzichte van het verticale vlak door AB een loodrechte stand heeft. De meetkundige situatie is dan volledig bepaald en er resteert slechts wat rekenwerk. In vectornotatie is BP gelijk aan

$$BP = p(AB - OZ \wedge AB \sin(\alpha) / \sin(\sigma))$$

waarbij $\wedge$ het symbool voor de vectorvermenigvuldiging is, en OZ een eenheidsvector in verticale richting. Deze formule vormt de kern van het desbetreffende programma PYT3DBT, dat overigens een bijna getrouwe kopie is van het programma PYT3BT.

In figuur 10.4 tonen we een voorbeeld van een met dat programma verkregen boom. Bij de projectie is gebruik gemaakt van de bovenstaande projectieformules. In het programma is gekozen voor een vaste links-rechtsinstelling en een door de gebruiker willekeurig te kiezen kijkhoek. Men kan met het programma de boom zowel van boven als van opzij beschouwen, en alles ertussenin.

Uiteraard kan het programma op vele manieren gevarieerd en uitgebreid worden. In figuur 10.3 zou de driehoek BPQ met behoud van de verkortingsverhoudingen p, q en de hoeken α, β een wat gedraaide stand kunnen innemen. Dat is vooral wenselijk wanneer een der hoeken erg klein is. Er treedt dan een soort monopodale groei op, waarbij de hoofdstam nagenoeg verticaal gericht blijft. Met het gegeven programma zou de te vormen boom een nogal platgeslagen vorm gaan aannemen. Om dat te vermijden dient er telkens een extra draai om de z-as, de zogenaamde divergentie, aan toegevoegd te worden.

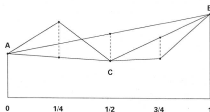


Figuur 10.4 Een Japanse boom van Pythagoras

Ons hoofddoel is het ontwerpen van een fractalachtig berglandschap, maar omdat dat een nogal ingewikkelde zaak is beperken we ons eerst tot wat eenvoudigers, namelijk het ontwerpen van een verticale, vlakke doorsnede door een synthetisch landschap. Het is zoiets als een schoolopgave, waarbij men op een kaart van Europa een lijn trekt van Keulen tot Parijs en men een grafiek moet maken van de hoogte ten opzichte van het zeeniveau. Mandelbrot heeft erop gewezen dat een dergelijke grafiek veel weg heeft van een toevallige fractal, zoals de lijn van Brown in hoofdstuk 7, en dat men de fractaldimensie van het profiel als een goede geografische karakteristiek kan gebruiken. Een dergelijk profiel kan gemakkelijk synthetisch worden gevormd met behulp van een methode die Richard Voss ook gebruikt heeft voor het ontwerpen van zijn spectaculaire landschappen, die men in de literatuur kan aantreffen. Het uitgangspunt is als in figuur 10.5 een lijnstuk dat we gemakshalve opvatten als een stuk van de getallenlijn van 0 tot 1. De constructie steunt op voortdurend halveren en het toekennen van hoogten volgens een stochastisch proces, dat toevallige getallen produceert met een gemiddelde van 0 en een strooiing van 1. De toevallige getallen dienen daarbij verdeeld te zijn volgens de normaalverdeling of iets wat erop lijkt.

We beginnen met de punten 0, 1, waaraan we een willekeurige hoogte toekennen, stochastische waarden bepaald door een normaalverdeling met gemiddelde 0 en strooiing σ_0. In het midden 1/2 wordt de hoogte primair vastgelegd als de gemiddelde waarde van die in de omliggende punten. Daarna wordt die waarde verhoogd of verlaagd met een toevallige waarde, bepaald door een normaalverdeling met gemiddelde 0 en een kleinere strooiing σ_1. In figuur 10.5 zien we dat meetkundig vertaald als de vervanging van de rechte lijn *AB* door de gebroken lijn *ACB*. Bij de volgende stap worden de middens van de lijnstukken *AC* en *CB* op hun beurt verticaal verplaatst over toevallige afstanden volgens dezelfde normaalverdeling, maar met een kleinere strooiing σ_2. In vakjargon heet de gevolgde methode de 'midpoint displacement'. Zo kunnen we in gedachten oneindig vaak doorgaan, maar voor een afbeelding op het beeldscherm is het niet zinvol verder te gaan dan een gebroken lijn die uit 512 kleine lijnstukjes bestaat.

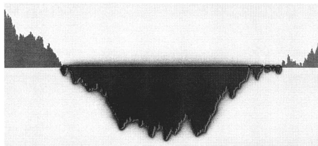
Om de gewenste zelfgelijkvormigheid te kunnen aanbrengen, dienen de opeenvolgende strooiingen volgens een meetkundige rij kleiner te worden. Uit de eigenschappen van een Brownse beweging kan men langs theoretische weg afleiden dat bij halvering van de lengteschaal een halvering van de variantie σ^2 past en dat de fractaldimensie van de corresponderende lijn, afgebeeld in figuur 1.2, gelijk is aan 1.5. Algemener kunnen we voor de rij van strooiingen een factor $(1/2)^H$ aannemen, waarbij *H* tussen 0 en 1 inligt. Dus



Figuur 10.5 De constructie van Voss

$\sigma_1 = (1/2)^H \sigma_0, \quad \sigma_2 = (1/2)^H \sigma_1$, enzovoort

De keuze van σ_0 doet er niet veel toe, men kan daarvoor 1 nemen. Ook weer theoretisch kan worden bewezen dat het resulterende profiel een fractal is met de dimensie $2 - H$. In figuur 10.6 is een dergelijk profiel, verkregen met behulp van het programma PROFILE, afgebeeld.
In de natuur komen dergelijke grillige lijnen veel voor, van een ECG-patroon tot de statistiek van oceaangolven. De rimpels op een wateroppervlak wijzen op een profiel met een hoge waarde van H, 0.92 bijvoorbeeld. Bij een meer ruisachtige grafiek behoort daarentegen een lage waarde van H, zoals 0.1. Ook hier kan men weer al experimenterende ervaring opdoen met de computer door het genoemde programma een aantal malen te herhalen met telkens een andere waarde voor H.

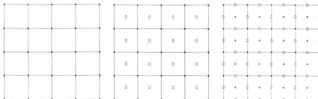


Figuur 10.6 Een fractalprofiel verkregen met de constructie van Voss

Een fractallandschap

De constructie van Voss kan gemakkelijk uitgebreid worden tot een overeenkomstige synthese in twee dimensies. Het uitgangspunt is nu een vierkant in het x,y-vlak met de hoekpunten (0,0), (1,0), (1,1), (0,1). Het beoogde landschap vormen we als een functie $z(x,y)$ die aan een willekeurig punt (x,y) een z-waarde als hoogte toevoegt. Het niveau $z = 0$ kunnen we desgewenst als de waterspiegel opvatten, zodat een negatieve waarde van z dan met water overeenkomt. In de hoekpunten van het grote vierkant kiezen we de hoogten weer op toevallige wijze als waarden uit een normaalverdeling met gemiddelde 1 en strooiing σ_0. Vervolgens bepalen we de hoogte in het centrum (1/2,1/2) als het gemiddelde van de hoogten van de vier hoekpunten, aangevuld (vertical displacement) met een toevallige waarde uit een normaalverdeling met een kleinere strooiing σ_1. Daarna bepalen we de hoogten van de middens van de zijden en zo verder. In figuur 10.7 tonen we drie opeenvolgende fases in deze procedure. Het blijkt dat we te maken hebben met afwisselend een horizontaal-verticaal patroon van vierkantjes en een diagonaal patroon van vierkantjes, waarbij er een verkleiningsfactor is van $1/\sqrt{2}$. De centra van de vierkantjes van het ene patroon worden dan de hoekpunten van het volgende patroon. Steeds wordt de hoogte in een nieuw punt berekend als het gemiddelde van de reeds berekende hoogten van de omliggende hoekpunten, en steeds wordt

Figuur 10.7 De constructie van een fractaloppervlak

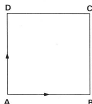


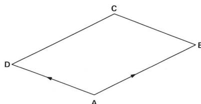
dat gemiddelde, meetkundig gesproken, verticaal verschoven met een toevallige grootheid van een normaalverdeling. Uiteraard dienen aan de rand van het grote vierkant speciale maatregelen genomen te worden, omdat daar in het algemeen slechts de hoogten van drie omliggende punten in aanmerking komen.

Van deze constructie bestaat een aantal varianten. Men kan bijvoorbeeld telkens nadat voor de hoekpunten van een nieuw kwadratisch rooster de hoogten bepaald zijn, ook aan de reeds berekende hoogten nog een toevallige verplaatsing aanbrengen. De fraaie, kleurrijke landschappen, achtergrond voor science ficton-films, die men in de computer graphics-literatuur kan aantreffen, zijn doorgaans het resultaat van krachtige, professionele computers, maar het is goed mogelijk ook thuis bevredigende resultaten te verkrijgen. De constructie van Voss kan men dan het beste uitvoeren in een vierkant dat in 64×64 of 128×128 pixels is verdeeld. Men kan dan de berekende hoogten opbergen in een dubbele array met even zovele rijen en kolommen. Er wordt daarbij wel een forse aanslag op het computergeheugen gedaan, maar wanneer men de hoogten uitdrukt in gehele getallen, kan men volstaan met een 'integer array' en lukt het met een (128,128)-matrix nog wel. Aldus is in het programma VOSSFS een dergelijke hoogtelijst gemaakt. Aan de basis van dat programma staat een willekeurige rij van random getallen, bepaald door een enkel geheel getal, de 'seed'. Voor elke seed wordt dus weer een ander fractaloppervlak berekend.

Coulissenprojectie

In het programma PROJS geven we een eenvoudige methode om van een door een wiskundige functie $z = f(x,y)$ bepaald oppervlak een ruimtelijke projectie te maken, waarbij wat voor het oog onzichtbaar is, ook niet wordt afgebeeld. Dat zelfde programma kan vrij gemakkelijk geschikt gemaakt worden om een oppervlak, een fractaloppervlak bijvoorbeeld, af te beelden dat door een hoogtetabel in de vorm van een matrix is bepaald. Het programma is geschreven voor een EGA- of een VGA-resolutie waarbij het beeldscherm opgedeeld is in 640 pixelkolommen. We stellen ons voor dat het oppervlak in ruimtelijke coördinaten x, y, z zich uitstrekt over het eenheidsvierkant $0 < x < 1$, $0 < y < 1$ en dat z in absolute waarde kleiner is dan 1. De beeldschermcoördinaten noemen we u, v waarbij $0 < u < 640$ en $-1 < v < 1$. Door een projectie als aan het begin van dit hoofdstuk beschreven wordt het x,y-vierkant op het beeldscherm geprojecteerd als een parallellogram $ABCD$, zoals aangegeven in figuur 10.8.





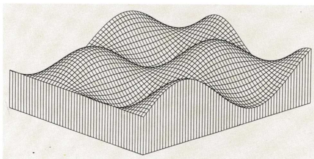
Figuur 10.8 De projectie van een vierkant

We bekommeren ons niet om de projectiehoeken α en β, maar we geven die hoekpunten een geschikte positie op het beeldscherm $A(288,-p)$, $B(608,q)$, $C(352,p)$ en $D(32,-q)$, waarbij $p=9q$. Met p hebben we de mogelijkheid het parallellogram wat smaller of breder te maken. In feite is dat hetzelfde als een verandering van de kijkhoek. Tussen A en B bevinden zich 320 pixelkolommen, en tussen A en D zijn er 256, bewust gekozen mooie getallen.
Het ruimtelijke oppervlak denken we doorsneden door twee series verticale vlakken, zogenaamde coulissen, $Nh+1$ in de x-richting en $Nv+1$ in de y-richting. Geschikte getallen zijn $Nh=40$ en $Nv=32$, omdat zij van 320 en 256 in dezelfde factor 8 verschillen. We letten eerst op de serie coulissen die evenwijdig aan de x-as zijn, en we bouwen ze van voren naar achteren op. Elke coulisse stellen we ons voor als een vlakke, ondoorzichtige, verticale plaat met de begrenzing $0<x<1$ en $-1<z<f(x,y)$. Op het beeldscherm zien we van de coulisse alleen de door $z=f(x,y)$ bepaalde begrenzing als de projectie van een gebogen lijn, voorzover die lijn niet afgedekt is door de projectie van een eerder geplaatste coulisse. Om te weten wat als reeds afgedekt en dus als onzichtbaar kan worden beschouwd, houden we een grens van zichtbaarheid bij in een array welke uit net zoveel elementen bestaat als er gebruikte pixelkolommen zijn, dus hier een array als $s(576)$, een soort maximum hoogtemeter. Komt bij de punt-voor-punt projectie van de bovengrens van een coulisse een bepaalde pixelkolom aan de beurt, dan moet de v-waarde van het af te beelden punt vergeleken worden met de in het bijpassende array-element voorhanden grenswaarde. Is v groter dan de grens, dan wordt het punt afgebeeld, maar wordt tegelijk de grens verhoogd tot het nieuwe niveau v. Is v kleiner dan de grens, dan wordt niets afgebeeld en gaan we verder naar de volgende kolom. Zijn alle coulissen aldus afgebeeld, dan komt op overeenkomstige wijze de serie coulissen evenwijdig aan de y-as aan de beurt. Ten slotte zien we op het beeldscherm het landschap op suggestieve wijze weergegeven als een netwerk van elkaar loodrecht doorkruisende lijnen.

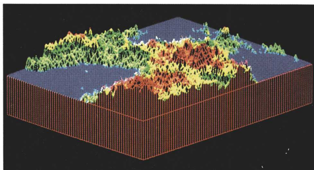
Landschapskunst

In figuur 10.9 tonen we aldus een golvend landschap bepaald door een functie van het type $z=\sin(ax)\cos(by)$, waarbij a en b geschikt gekozen getallen zijn.
Wanneer we de constructie van Voss combineren met de beschreven projectiemethode, dan krijgen we een fractallandschap als afgebeeld in figuur 10.10.

Figuur 10.9 Een golvend landschap



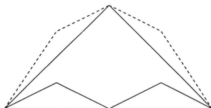
Figuur 10.10 Een fractallandschap in coulissenprojectie



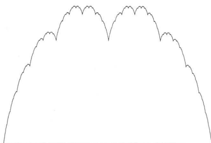
In deze afbeelding zien we een soort kustlandschap met een stuk zee waarin zich eilandjes bevinden, iets als Japan. In grote lijnen is dat landschap op de volgende wijze ontstaan. Eerst wordt volgens de methode van Voss een integer array $Z(128,128)$ gemaakt waarin de hoogten op een puntenrooster van 129×129 bevat zijn als positieve of negatieve gehele getallen. Het rooster denken we uitgespreid over het eenheidsvierkant $0 < x < 1$, $0 < y < 1$ en de hoogten reduceren we tot het interval $(-1,1)$. Vervolgens vatten we alles wat beneden een bepaald niveau is, bijvoorbeeld $z = 0$, op als water. Bij de projectie moeten we kunnen beschikken over de gereduceerde hoogte op een willekeurige x,y-positie. In het algemeen ligt die positie tussen twee of vier roosterpunten in, en dan leiden we de verlangde hoogte eenvoudig af uit die van de omringende roosterpunten door lineaire interpolatie.

De lijn van Van der Waerden

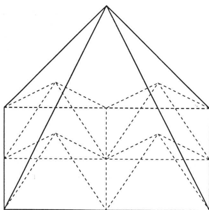
Tegenover de op toeval berustende fractallandschappen staan de deterministische fractaloppervlakten waaraan een soms heel eenvoudige wiskundige constructie ten grondslag ligt. Bij wijze van aanloopje beginnen we, eenvoudiger, met een fractallijn in twee dimensies, een kromme in 1930



Figuur 10.11 De constructie van de lijn van Van der Waerden



Figuur 10.12 De fractal van Van der Waerden



Figuur 10.13 Een piramide als begin van de Takagi-berg

bedacht door een beroemde landgenoot, B. L. van der Waerden. Als geschetst in figuur 10.11 gaan we uit van een op het eenheidslijnstuk (0,1) geplaatste dakkantfunctie of tent

$$f(x) = x \quad \text{als } 0 < x \leqslant 1 < 2$$
$$f(x) = 1 - x \quad \text{als } 1/2 < x \leqslant 1$$

Die functie denken we periodiek herhaald met de periode 1, dus

$$f(x) = f(x - \text{int}(x))$$

voor willekeurige waarden van x.
We vormen nu stap voor stap de bedoelde fractallijn door bij wat reeds voorhanden is een verkleinde kopie van de tentfunctie op te tellen. De verkleiningsfactor is $1/2$ in de x-richting en a, een willekeurig getal, in de y-richting. De eerste paar bijdragen zijn eveneens in figuur 10.11 aangegeven. Het proces laat zich ook op eenvoudige wijze in formulevorm brengen als

$$\text{F}(x) = f(x) + af(2x) + a^2 f(4x)/4 + a^3 f(8x)/8 + \ldots$$

of in recursieve, en nog eenvoudiger, vorm

$$\text{F}(x) = a\text{F}(2x) + f(x)$$

Voor Van der Waerden was deze kromme, afgebeeld in figuur 10.12 met behulp van het programma WAERDEN, een eenvoudig voorbeeld van een functie die wel continu, maar niet differentieerbaar was. Later bleek dat de Japanner Takagi reeds in 1903 hetzelfde had bedacht. Oorspronkelijk was bij hen de verticale verkorting gelijk aan de horizontale, maar met een waarde van a tussen $1/2$ en 1 in kunnen we een fractal krijgen waarvan de dimensie tussen 1 en 2 inligt, in formulevorm $1 + \log(2a)/\log(2)$. Bij de speciale keuze $a = 1/2$ van Van der Waerden is de fractaldimensie overigens precies 1.
De constructie kan op verschillende wijzen gevarieerd en gegeneraliseerd worden. In plaats van de tentfunctie zou men net zo goed een andere periodieke functie kunnen kiezen, bijvoorbeeld $f(x) = \text{abs}(\sin(\pi x))$. Maar liever dan uit te weiden over dit soort uitbreidingen gaan we de driedimensionale variant bekijken.

De berg van Takagi

Bij de ruimtelijke versie bouwen we een berg op het eenheidsvierkant $0<x<1$, $0<y<1$ door uit te gaan van een eenvoudig, periodiek beginoppervlak $f(x,y)$. Eenvoudige voorbeelden zijn de piramidale tent (zie figuur 10.13)

$$f(x,y) = \min(1/2 - \text{abs}(x - 1/2), 1/2 - \text{abs}(y - 1/2))$$

de sinusberg

$$f(x,y) = \sin(\pi x)\sin(\pi y)$$

of

$$f(x,y) = xy(1 - x)(1 - y)$$

Al deze definities gelden primair binnen het eenheidsvierkant. Daarbuiten moeten x en y ter wille van de gestelde periodiciteit vervangen worden door het fractionele deel van x en y, of door $x - \text{int}(x)$ en $y - \text{int}(y)$. De uiteindelijke hoogte $F(x,y)$ van de fractalberg wordt dan gedefinieerd als de reeks

$$F(x,y) = f(x,y) + af(2x,2y) + a^2f(4x,4y) + a^3f(8x,8y) + \ldots$$

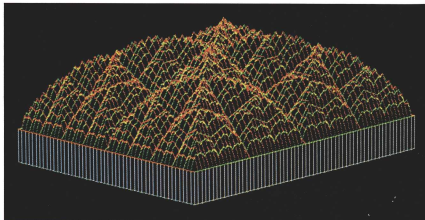
waarbij de factor a gelijk aan $1/2$ kan zijn. In figuur 10.14 geven we een volgens deze methode via het programma TAKAGI geconstrueerde berg.

Een hoogtekaart

Wanneer we ons voorbereiden op een vakantie, een trektocht in een berglandschap bijvoorbeeld, of een zeiltocht over vaart, meer en zee, plegen we ons te bedienen van kaarten die een indruk geven van wat ons te wachten staat. Voor een bergwandeling kan een reliëfkaart, een panoramische projectie, heel nuttig zijn, maar bij een boottocht zijn we meer geïnteresseerd in diepteverschillen. Trouwens ook in een gewone atlas treffen we vele kaarten aan die ons met kleurverschillen een indruk geven van hoogte en diepte. Het komt erop neer dat we van het landschap een verticale projectie, een bovenaanzicht, maken en dat we hoogte of diepte weergeven als kleurgradaties. De kleur is met andere woorden een maat voor de derde dimensie. Voor een computer met de gebruikelijke zestien EGA- of VGA-kleuren is dat een ideale manier. In beginsel kan men die kleuren vrij kiezen uit een beschikbaar aantal van 64 of meer, en kan men met geschikte software de kleuren als een verlopende schaal van bijvoorbeeld blauwe tinten voor water en bruin voor de bergen kiezen. Er ligt hier weer een heel terrein braak voor eigen experimenten. Deze projectiemethode is zowel eenvoudig als snel en bijzonder geschikt voor een eenvoudige microcomputer. In het programma FRSURF laten we zien hoe het werkt. We hebben daarin gekozen voor de Takagi-berg, samengesteld uit een reeks

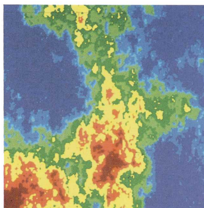
piramiden. Op het beeldscherm kiezen we een vierkant $0<i<n$, $0<j<n$ van pixels en dat laten we overeenstemmen met het eenheidsvierkant $0<x<1$, $0<y<1$ waarin de bergfunctie $F(x,y)$ gedefinieerd is. De positie (i,j) van het beeldscherm correspondeert dan met het niveau $z=F(i/n,j/n)$. Die hoogte kunnen we met een willekeurig aan te passen schaalfactor c vertalen in een geheel getal $\text{col}=\text{int}(cz)$ als kleurindex. Vinden we dat bijvoorbeeld acht kleuren voldoende zijn, dan kunnen we het kleurgetal modulo 8 reduceren. Omdat in de gegeven situatie de bergfractal symmetrisch is ten opzichte van horizontale, verticale en diagonale

Figuur 10.14 De Takagi-berg





Figuur 10.15 Een hoogtekaart van de Takagi-berg



Figuur 10.16 een topografische fractal

richtingen is het voldoende de waarde van *F* slechts in een octant van het eenheidsvierkant echt uit te rekenen. Op die manier kunnen we in het algemeen telkens acht punten tegelijk op het beeldscherm afbeelden. Het resultaat van een willekeurige keuze zien we in figuur 10.15.
Het programma FRSURF is even eenvoudig als veelzijdig. Omdat de berekening van $F(x,y)$ in een subroutine is opgenomen, kan in principe voor elke andere functie gemakkelijk een dergelijke hoogtekaart gemaakt worden. Men lette er alleen op dat voor functies met minder symmetrieën het programma een weinig aangepast dient te worden.

Een atlas van fractals

We zijn nu in staat om een willekeurige reeks van fractalachtige landkaarten te ontwerpen met geheimzinnige zeeën, onbekende eilandrijken en fantasievolle bergen. Van de benodigde software zijn de essentiële elementen hierboven al besproken, al komt er bij de uitwerking van de programma's nog wel wat detailwerk kijken. Het volledige programma is, met andere programma's, op aanvraag beschikbaar. Voor elke random 'seed' kan met behulp van de methode van Voss van het te maken landschap een hoogtetabel gemaakt worden in een array van 128 × 128 roosterpunten. In figuur 10.10 hebben we een dergelijk landschap in scheve projectie afgebeeld volgens de coulissenmethode. Maar nu kan dezelfde tabel dienen als grondslag van een topografische kaart. Op het beeldscherm is een vierkant van 128 × 128 pixels wat mager en daarom hebben we het vierkant uitgebreid tot 256 × 256 pixels, waarbij de ontbrekende hoogten door interpolatie uit die van de omringende punten zijn afgeleid. Alles wat lager is dan een vooraf ingesteld zeeniveau geven we een donkerblauwe kleur en vervolgens gebruiken we telkens hoger gaande, als in een echte atlas, de kleuren lichtblauw, lichtgroen, donkergroen, geel enzovoort. Het resultaat tonen we in figuur 10.16.
Met dezelfde methode kunnen ook synthetische wolkenluchten verkregen worden. Alleen de kleuren moeten aangepast worden, van donkerblauw tot geelwit bijvoorbeeld.

Julia sets I

De Julia sets vormen het hoogtepunt van de wiskundige fractals. De wiskundige definitie berust op een eenvoudig principe en het is werkelijk een wonder welk een rijkdom aan meetkundige structuren er verborgen ligt in een combinatie van slechts enkele eenvoudige, wiskundige bewerkingen. Dat wonder kan men, mits toegerust met de nodige wiskundekennis, enigszins begrijpen en verklaren, maar toch staan ook ervaren wiskundigen nog vaak voor raadsels, en vormen de Julia sets nog steeds een actief beoefend wiskundig thema. Wie als niet-wiskundige, alleen gewapend met algemene schoolse kennis, toch iets meer wil dan alleen naar mooie plaatjes kijken, al of niet met nauwelijks begrepen programma's, dient in elk geval enigszins vertrouwd te zijn met de rekenwijze van de complexe getallen. Ten gerieve van de lezer is datgene wat daarvan beslist nodig is opgenomen in appendix 1. Daarnaast is enige kennis nodig van sinus en cosinus, en moet men van enkele eenvoudige functies de afgeleide of differentiaalquotiënt kunnen bepalen. Van de meetkunde hebben we vooral te maken met de in hoofdstuk 3 besproken gelijkvormigheidstransformaties. Het verband tussen 'rekenen' en 'meten' wordt gelegd door het x,y-coördinatenstelsel. Om een goed inzicht te krijgen in de eigenschappen van Julia sets is heel wat kennis van wiskunde nodig, veel meer dan we op school plegen te verwerven. Maar toch kan zonder hogere wiskunde zoveel interessants gezegd worden, dat het onderwerp verdeeld moest worden over twee hoofdstukken. In dit hoofdstuk komen hoofdzakelijk de Julia sets aan de orde die bepaald zijn door een veelterm van de tweede graad, de zogenaamde kwadratische familie.
Bij de Julia sets komen veel eerder besproken zaken bij elkaar. Een Julia set is nauw verbonden met een iteratief systeem, een dynamisch systeem van complexe getallen of punten van de vorm

(11.1)
$z \rightarrow f(z)$

In zekere zin is de Julia set de verzameling van alle banen die zich niet normaal gedragen. Normaal gaat een baan ergens heen, is er een aantrekkend punt of, algemener, een aantrekkende verzameling waar banen naartoe gaan. Populair gesproken zijn er speciale banen die niet weten waar ze heen zullen gaan, banen van eeuwige twijfelaars, en die behoren tot de Julia set. De Julia set zou men kunnen opvatten als een soort waterscheiding. Punten die een klein beetje naast de Julia set liggen, hebben wel een

normale baan. De baanpunten vluchten als het ware weg van de Julia set, die daarmee het karakter heeft van een 'strange repeller', een 'vreemde afstoter'.

De kwadratische familie

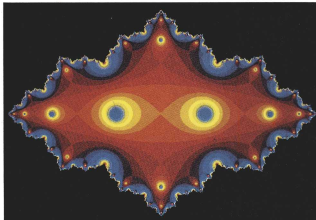
In dit hoofdstuk beperken we ons hoofdzakelijk tot kwadratische systemen, waarbij dus $f(z)$ een veelterm van de tweede graad is, dat wil zeggen van de vorm $f(z) = az^2 + bz + c$. Voor dit systeem behoort bij elk punt een opvolger, maar omgekeerd heeft bijna elk punt twee voorgangers, omdat een vierkantsvergelijking in het complexe getallensysteem altijd twee, eventueel samenvallende, wortels heeft. Er bestaat dan ook een bijzonder punt, een zogenaamd kritiek punt, dat slechts één voorganger heeft. Voor een willekeurige afbeelding $z \rightarrow f(z)$ is elk punt waarvoor de afgeleide verdwijnt, dus $f'(z) = 0$, een kritiek punt. De opvolgers van een zeker beginpunt vormen een enkelvoudige baan, maar alle voorgangers vormen met elkaar de structuur van een tweetallige boom.
Het kwadratische systeem (11.1) bezit twee dekpunten, die berekend kunnen worden als de twee wortels van de vierkantsvergelijking $f(z) = z$. Daarnaast moeten we ook $z = \infty$ als dekpunt meetellen. De functie $f(z)$ kan in het algemeen nog wat vereenvoudigd worden door het dynamische systeem te onderwerpen aan een gelijkvormigheidstransformatie. In meetkundige taal komt dat neer op een verschuiving, een draaiing en een schaling van de Julia set. Men kan aldus aan een of meer van de bijzondere punten een speciale ligging toekennen. In het bijzonder werken we bij het kwadratische systeem meestal met de volgende twee standaardvormen. In de eerste plaats

(11.2)
$z \rightarrow z^2 + c$

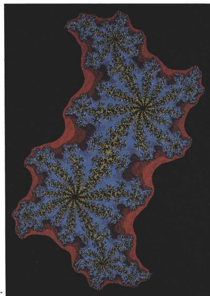
en vervolgens

(11.3)
$z \rightarrow \mu z + (1 - \mu)z^2$

Bij (11.2) is het kritieke punt $z = 0$ en dat kan in bepaalde situaties van groot praktisch en theoretisch nut zijn. De eindige dekpunten $1/2 \pm \sqrt{(1/4 - c)}$ zijn daarbij wat gecompliceerder. Bij (11.3) hebben de eindige dekpunten een speciale plaats in $z = 0$ en $z = 1$ gekregen. In de nog te bespreken programma's wordt de eerste vorm (11.2) het meeste gebruikt, met name in het kleurige pixelprogramma JUL1A. De tweede vorm (11.3) wordt toegepast in het overeenkomstige programma JUL1B. In beide programma's wordt de eigenlijke Julia set, een verzameling losse punten of verbonden lijnen, omgeven door kleurige banden.
De in dit hoofdstuk opgenomen kleurenillustraties zijn met deze programma's verkregen (figuur 11.1).



Figuur 11.1 De Julia set van $c = -0.55$



Figuur 11.2 Een Julia set waarvan een dekpunt een eigenwaarde met $r = 1.02$ en $\mu = 8\pi / 11$ heeft



Een spiraal van gebladerte

Dekpunten

De aard van een Julia set wordt in belangrijke mate bepaald door de aantrekkende dekpunten. Is z_0 een eindig dekpunt, dan is dus $f(z_0) = z_0$. In micro ziet $z' = f(z)$ in de omgeving van het dekpunt eruit als een gelijkvormigheidstransformatie

$$z' - z_0 = \mu(z - z_0)$$

waarbij

(11.4)
$$\mu = f'(z_0)$$

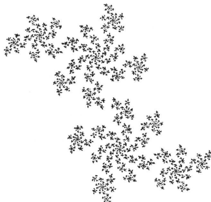
de zogenaamde *eigenwaarde* van het dekpunt.
We schrijven absolute waarde en argument van μ meestal als r en φ. Dat betekent dat we μ op de volgende wijze in reëel en imaginair deel kunnen splitsen

$$\mu = r(\cos(\varphi) + i\sin(\varphi))$$

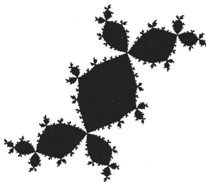
Voor (11.3) heeft μ dus reeds de betekenis van de eigenwaarde van het dekpunt $z = 0$. De eigenwaarde van het dekpunt $z = 1$ blijkt $2 - \mu$ te zijn. Bij de vorm (11.1) is de eigenwaarde $2z_0$ wanneer het dekpunt z_0 is.
Wanneer $0 < r < 1$, dan is het dekpunt aantrekkend en worden in elk geval de banen van punten die zich niet te ver van z_0 bevinden daardoor aangetrokken. De banen bewegen zich in het algemeen spiraalvormig naar het dekpunt toe. Alle punten die door z_0 worden aangetrokken, vormen met elkaar het aantrekkend gebied van dat dekpunt. Dat gebied kan bestaan uit een enkele samenhangende schijf of uit oneindig veel schijfjes (zie figuur 11.2).
In het bijzondere geval dat $\mu = 0$, heet het dekpunt superaantrekkend. Banen die tot z_0 convergeren doen dit zo sterk, dat het moeilijk is zich daarvan een goede voorstelling te vormen. Wellicht helpt de volgende analogie waarbij het gaat om een stapsgewijze benadering van π. Gewone aantrekking wil dan zeggen dat er bij elke stap een vast aantal decimalen van π wordt bijgevonden. Is dat aantal tien, dan zijn er toch nog een honderdduizend stappen nodig om een miljoen decimalen van π te kunnen berekenen. Superaantrekking betekent dat er bij elke stap evenveel correcte decimalen bijkomen als men reeds berekend had. Beginnen we bij een nauwkeurigheid van tien decimalen, dan is na twintig (!) stappen dat aantal ruimschoots bereikt. Men kan het zelf narekenen:

10,20,40,80,160,320,640,1280,2560,5120,$\approx$10000 enzovoort

Wanneer $r > 1$, dan is het dekpunt afstotend en maakt het volgens de theorie van Julia deel uit van de Julia set, een voor de praktijk nuttig gegeven (zie figuur 11.3).
Wanneer $r = 1$, dan heet het dekpunt *neutraal* of *indifferent*. Er kunnen



Figuur 11.3 Een onsamenhangende Julia set met $a = 0.11$ en $b = 0.66$



Figuur 11.6 Het konijn van Douady

dan tal van bijzondere verschijnselen optreden. Wie een mooie of interessante Julia set op het beeldscherm wil krijgen, dient zijn keuze van parameters op een dergelijke situatie af te stemmen. Is μ een zogenaamde echte eenheidswortel, is dus de verhouding φ/π een rationaal getal, bijvoorbeeld $2/5$, dan behoort het dekpunt tot de Julia set. Bij de Julia set zien we dan een soort bloembladachtige structuur, waarbij in het dekpunt en alle voorgangers daarvan telkens m blaadjes bij elkaar komen. Is $\arg(\mu)/\pi$ geen rationaal getal maar onmeetbaar, het gulden snede-getal $(-1+\sqrt{5})/2$ bijvoorbeeld, dan kunnen er nog vreemder zaken optreden, zoals zogenaamde Siegel-schijven, genoemd naar een wiskundige die dit verschijnsel beschreven en verklaard heeft (zie figuur 11.4). Een Siegel-schijf is gekenmerkt door een stelsel van concentrische banen. In het centrum bevindt zich een dekpunt of een voorganger van een dekpunt. In de praktijk kunnen we bij de kleuring letten op de gemiddelde afstand van de baanpunten tot het centrum, of op de hoek of fase van de baanpunten (zie figuur 11.5). De besproken classificatie van dekpunten is uiteraard ook geldig voor Julia sets van willekeurige functies $f(z)$. Er kunnen dan nog wel een paar extra bijzonderheden optreden. Zo is het mogelijk dat een dekpunt superstabiel is met een hogere exponent, bijvoorbeeld $f(z) = 2z^3 - z^4$. Het dekpunt $z = 0$ is daarbij superstabiel met exponent 3. Daarnaast zijn er de drie dekpunten 1, $(1 \pm \sqrt{5})/2$ welke alle afstotend zijn.

Periodieke cycli

Bij elke afbeelding $z \to f(z)$ behoort een hele familie van geïtereerde afbeeldingen die uit $f(z)$ ontstaan door herhaling. We kunnen ze beschrijven door $z \to f(f(z))$ of korter $z \to ff(z)$, en vervolgens $z \to fff(z)$ en zo verder. Een dekpunt van $z \to ff(z)$ is niet noodzakelijk een dekpunt van $z \to f(z)$, maar het kan wel aanleiding geven tot een periodieke cyclus van twee ele-

menten. Kiezen we bijvoorbeeld $f = z^2 - 1$, dan kunnen we constateren dat er de twee dekpunten $(1 \pm \sqrt{5})/2$ zijn en dat $-1,0$ een 2-cyclus is. De herhaalde afbeelding is $ff = z^4 - 2z^2$ en daarvan zijn zowel $(1 \pm \sqrt{5})/2$ als -1 en 0 gewone dekpunten.
In het algemeen is elk element van een periodieke m-cyclus van f een gewoon dekpunt van de m-voudig herhaalde afbeelding. Er is dus geen aparte theorie nodig van periodieke cycli. Net als gewone dekpunten kunnen cycli aantrekkend, afstotend en neutraal zijn. De eigenwaarde van een cyclus, het complexe getal dat de aard van de cyclus bepaalt, is gewoon de eigenwaarde van een van zijn elementen ten opzichte van de herhaalde afbeelding. Voor de praktijk is het meestal voldoende om in staat te zijn die eigenwaarde uit te kunnen rekenen voor een 2- of een 3-cyclus. Is voor de afbeelding $f(z)$ het paar z_1, z_2 een periodieke 2-cyclus, dan is de eigenwaarde μ gelijk aan

(11.7)
$$\mu = f'(z_1)f'(z_2)$$

Voor een periodieke 3-cyclus z_1, z_2, z_3 is de overeenkomstige formule

(11.8)
$$\mu = f'(z_1)f'(z_2)f'(z_3)$$

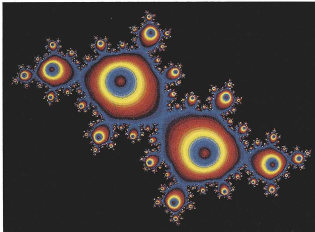
In het genoemde voorbeeld $f = z^2 - 1$ kunnen we dan vaststellen dat de 2-cyclus $-1,0$ de eigenwaarde 0 heeft en daarmee superstabiel is. Omdat een produkt 0 is zodra maar een van de factoren 0 is, komen we tot de conclusie dat een periodieke cyclus die een kritiek punt als element bevat superstabiel is. Voor de standaardafbeelding (11.2) is $z = 0$ een kritiek punt. Een periodiek superstabiele cyclus met $z = 0$ als element kunnen we dan op eenvoudige wijze maken door van de baan van $z = 0$ de eerstvolgende elementen op te schrijven

$$0, \quad c, \quad c^2 + c, \quad c^4 + 2c^3 + c^2 + c, \ldots$$

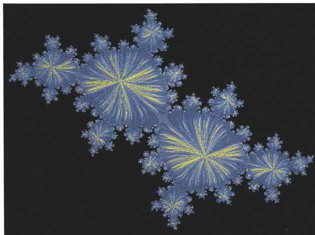
De eenvoudigste mogelijkheid is $c^2 + c = 0$ waaruit $c = -1$ volgt. We hebben dan de zojuist beschouwde superstabiele 2-cyclus. De volgende mogelijkheid is $c^4 + 2c^3 + c^2 + c = 0$. Afgezien van $c = 0$ volgt hieruit $c = -1.754878$ en $c = -0.122561 \pm 0.744862i$. Voor die speciale waarden van c kunnen we bij de Julia set iets bijzonders verwachten. Die zelfde waarden komen we ook tegen bij de analyse van de Mandelbrot set. De waarde $c = -0.122561 + 0.744862i$ is door Douady, een Frans wiskundige die zich veel met de kwadratische Julia set heeft beziggehouden, gekozen. De vorm doet sommigen aan een konijn denken en sindsdien staat de figuur bekend als 'het konijn van Douady' (figuur 11.6).

De theorie van Julia

Omstreeks 1919 heeft de Franse wiskundige en oorlogsinvalide Gaston Julia de grondslagen gelegd voor wat we nu Julia-fractals of Julia sets zijn



Figuur 11.4 Een Julia set met Siegel-schijven



Figuur 11.5 Dezelfde Julia set met fasekleuring

gaan noemen. Zijn theorie heeft betrekking op dynamische systemen van de vorm (11.1), waarbij f een quotiënt van twee veeltermen $P(z)$ en $Q(z)$ is, maar zoals gezegd beperken we ons hier in hoofdzaak tot het speciale geval (11.2) of (11.3). De door een functie f bepaalde Julia set noteren we als $J(f)$. De belangrijkste resultaten kunnen we als volgt samenvatten.

1 De Julia set kan gedefinieerd worden als een limietverzameling waarvan alle afstotende dekpunten en periodieke cycli deel uitmaken.
2 Behoort een punt P tot $J(f)$, dan behoren ook alle opvolgers en alle voorgangers van P tot $J(f)$.

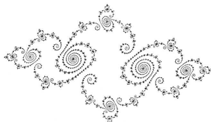
3 $J(f)$ is een aantrekker voor het inverse dynamische systeem $f(z) \rightarrow z$.
4 Behoort een punt P tot $J(f)$, dan wordt $J(f)$ volledig overdekt door de voorgangers van P.
5 Behoort een punt P tot $J(f)$ en is P geen element van een periodieke cyclus, dan is de baan van P chaotisch.
6 Is $f(z)$ een veelterm in z, dan is $J(f)$ of geheel onsamenhangend, een stoffractal, of met boogjes samenhangend, een lijnfractal. Is $f(z)$ een quotiënt van twee veeltermen, dan kan $J(f)$ uitzonderingsgewijs uit het hele vlak bestaan.
7 Is $f(z)$ een veelterm, dan kan $J(f)$ opgevat worden als de rand van het aantrekkende gebied van het oneindig verre dekpunt.

Deze eigenschappen kunnen ons bij de constructie van een Julia set grote diensten bewijzen. Het is blijkbaar al voldoende om één punt van de Julia set te kennen om de hele set in handen te krijgen. Meestal kan men met wat eenvoudig gereken wel een enkel afstotend dekpunt bepalen, en als dat niet lukt is er geen nood. Men neemt gewoon een willekeurig punt en men bepaalt daarvan een rij voorgangers, telkens een stap in het dynamische proces $z \rightarrow f(z)$ teruggaand. Na zeg een twintig stappen is men al zo dicht tot de Julia set genaderd dat, gezien het oplossend vermogen van computer of plotter, het laatst verkregen punt wel tot $J(f)$ gerekend mag worden. Maken we in het kwadratische geval gebruik van (11.3), dan is er helemaal geen probleem. Er zijn de twee dekpunten 0 en 1. Een eenvoudige overweging laat zien dat ze niet tegelijk aantrekkend kunnen zijn, zodat een van de twee zeker afstotend is en daarmee tot $J(f)$ behoort.

Het stabiele gebied

De banen van punten op de Julia set $J(f)$ kunnen we instabiel noemen. De Julia set is een afstoter, en zodra in de praktijk een berekend baanpunt een weinig afwijkt van zijn theoretische positie op de Julia set, wordt die afwijking bij volgende baanpunten steeds groter. De banen convergeren dan in het algemeen tot een enkel aantrekkend punt of een aantrekkende cyclus. Het theoretisch chaotische gedrag op $J(f)$ maakt dan plaats voor een stabiel gedrag. Een heel eenvoudig voorbeeld is het systeem $z \rightarrow z^2$. De Julia set is dan de eenheidscirkel $\text{abs}(z) = 1$, een van de zeldzame gevallen dat $J(f)$ geen echte fractal is. Een baan die ook maar heel weinig buiten de cirkelomtrek start, loopt na een gering aantal stappen snel weg naar de aantrekker in het oneindige. Een baan die iets binnen de cirkelomtrek begint, gaat na een paar stappen snel naar de oorsprong $z = 0$ toe. In dit voorbeeld onderscheiden we een instabiel gebied als de Julia set en een stabiel gebied als het binnengebied en het buitengebied van de cirkel. In het algemeen is het stabiele gebied gewoon het complement van de Julia set, alles wat niet tot $J(f)$ behoort.
Pas in de laatste jaren is er volledige klaarheid gekomen in de structuur van het stabiele gebied. In eerste instantie bestaat het stabiele gebied uit de aantrekkende gebieden van de aantrekkende dekpunten en aantrekkende periodieke cycli. Een aantrekkend gebied van een enkel dekpunt moeten we ons in het algemeen voorstellen als of een enkel schijfvormig

gebied, of een oneindige reeks van schijfjes waarvan er één, de hoofdschijf, de aantrekker bevat. De andere schijfjes zijn daarvan de voorgangers. Door de transformatie *f* worden in zo'n serie schijfjes een positie opgeschoven, maar wordt de hoofdschijf in zichzelf afgebeeld. Voor het aantrekkende gebied van een periodieke *m*-cyclus geldt iets soortgelijks. In het hoofdgebied worden de schijfjes, verbonden met de *m* elementen van de cyclus, dan periodiek verwisseld. Daarnaast kunnen er schijfjes in het stabiele gebied zijn waarbij dekpunten of cycli neutraal zijn. In dat geval kunnen bijvoorbeeld de reeds besproken Siegel-schijven optreden. Ten slotte kunnen er, heel uitzonderlijk, nog zogenaamde Herman-ringen voorkomen, maar in het kwadratische geval komen die niet voor (figuur 11.7).



Figuur 11.7 De Julia set van $a = -0.7543$ en $b = 0.1130$

Praktijk

De beste manier om kennis te maken met de Julia-fractals is om te experimenteren met een van de standaardprogramma's JULIAMC, JULIABT, JULFILL of JUL1A. Men heeft daarbij de keuze uit een aantal mogelijkheden. Men kan alleen een afbeelding van *J(f)* als stofwolk of als spel van lijntjes maken, en daarbij kunnen kleuren gemist worden. Desgewenst kan een samenhangende Julia set ook gevuld worden door alleen het binnengebied te markeren (met JULFILL).

Mooier is het om *J(f)* van buiten en/of van binnen te kleuren met gekleurde banden (met JUL1A enzovoort). De buitenkant is het eenvoudigst en daartoe is er een eenvoudig recept. We weten dat de baan van een punt *z*, dat zich in het buitengebied bevindt, door het oneindig verre dekpunt wordt aangetrokken. De mate waarin dat geschiedt kunnen we kwantificeren door te tellen na hoeveel iteratiestappen de baan voldoende ver weg is. Wat voldoende ver weg is of, anders gezegd, wat voldoende dicht bij oneindig is, kunnen we zelf afspreken, bijvoorbeeld verder weg dan een cirkel met straal 100 of zoiets. Bij elk punt *z* in het buitengebied behoort dus een soort vluchtgetal $m(z)$, dat zegt dat op de *m*-de iteratiestap het baanpunt voor het eerst zich verder van $z = 0$ verwijderd heeft dan een bepaalde vooraf ingestelde afstand, zeg 100. Dat vluchtgetal kunnen we in het computerprogramma omzetten in een kleurwaarde. Om desgewenst ook het binnengebied te kunnen kleuren kan men gebruik maken van een overeenkomstig aantrekkingsgetal, bijvoorbeeld bij aanwezigheid van een aantrekkend dekpunt. Men spreekt een klein getal af, zeg 0.01, als straal van een cirkelvormige omgeving van dat dekpunt. Zodra een punt van een door het dekpunt aangetrokken baan voor het eerst in die omgeving terechtkomt, wordt het desbetreffende rangnummer gebruikt voor de kleurwaarde.

De pixelmethode

Wie dom is moet hard werken, en dat kost tijd, een regel die ook in de computerwereld geldt. Wie een Julia set op het beeldscherm wil aanschouwen, ziet zich genoodzaakt een daartoe strekkend programma te schrijven. Dat kan heel goed op de volgende gemakzuchtige manier. We denken de Julia set omgeven door een rechthoek en laten die op het beeldscherm corresponderen met een rechthoek van pixels, 400×300 bijvoorbeeld. Voor elk van die 120 000 punten bepalen we een vluchtgetal. Houden we voorlopig de afbeelding $z \rightarrow z^2 + c$ aan als standaardkeuze, dan moeten we voor elk beginpunt x,y een aantal baanpunten berekenen volgens $z_1 = f(z)$, of gesplitst in reëel en imaginair deel

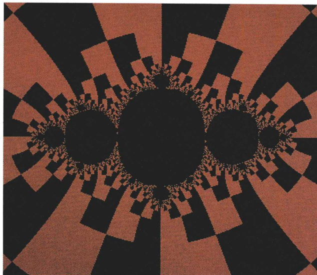
(11.9)

$x_1 = x^2 - y^2 + a$

$y_1 = 2xy + b$

Willen we het buitengebied van de Julia set kleuren, dan moeten we toetsen voor welk rangnummer het baanpunt voldoende ver van de oorsprong is gekomen. Telkens wanneer we een nieuw baanpunt berekend hebben, moeten we dus ook de kwadraatafstand $x^2 + y^2$ berekenen en kijken of die groter is dan bijvoorbeeld 10 000. Om enigszins nauwkeurig te kunnen werken, moeten we zonodig veel baanpunten van een zelfde baan kunnen onderzoeken, zeg een honderdtal. Dat betekent dat we vele miljoenen keren een zelfde berekening moeten uitvoeren. Een dom werkje, waar een computer niet tegenop ziet, maar dat wel veel tijd kost. Deze methode is beschreven in het programma JUL1A, een programma dat eigenlijk alleen goed uitvoerbaar is op een moderne computer met liefst zestien EGA-kleuren. Met kleine slimmigheden kan het programma wel wat versneld worden. Omdat de Julia set in zijn totaliteit altijd symmetrisch ten opzichte van de oorsprong is, behoeven we van de pixelrechthoek alleen de helft te onderzoeken. Is bovendien c reëel, dat wil zeggen is $b = 0$, dan is de Julia set zowel horizontaal als verticaal symmetrisch en kunnen we met een kwart van de pixelrechthoek volstaan. Natuurlijk kunnen we heel bescheiden met een kleine rechthoek, zeg van 80×60 pixels, beginnen. De winst in tijd ten opzichte van de grote rechthoek is dan al een factor 25. Willen we toch een grote afbeelding die liefst schermvullend is, dan kunnen we altijd nog de computer laten werken terwijl we zelf wat anders doen.
Met behulp van het programma JULFILL kan het binnengebied van een Julia set, mits aanwezig, gemarkeerd worden. Kleuren zijn dan niet nodig. Voor een stofachtige Julia set heeft dat geen zin, omdat alle banen of naar oneindig gaan, of tot de Julia set zelf behoren. De waarden van c waarvoor J lijnsamenhangend is en waarvoor een inwendige is, corresponderen met het binnengebied van de Mandelbrot set die in hoofdstuk 13 wordt besproken.
Willen we ook het binnengebied van de Julia set kleuren, dan moeten we de banen toetsen op convergentie tot een aantrekkend dekpunt of tot een aantrekkende periodieke cyclus. Is er een aantrekkend dekpunt x_0,y_0 bekend, dan kunnen we natuurlijk kijken of de kwadraatafstand $(x - x_0)^2 +$

$(y - y_0)^2$ voldoende klein is. Is er wel een aantrekkend dekpunt maar kennen we de coördinaten niet, dan kunnen we toetsen op de nabijheid van twee opvolgende baanpunten. Is x_1,y_1 de opvolger van x,y, dan kunnen we toetsen of de kwadraatafstand $(x_1 - x)^2 + (y_1 - y)^2$ voldoende klein is. In het programma JUL1A, een kleurenvariant van JULFILL, is dit nader uitgewerkt. Voor een situatie met een aantrekkende 2-cyclus moet de toets natuurlijk aangepast worden, bijvoorbeeld door x,y te vergelijken met de opvolger van x_1,y_1. We slaan daarbij dus telkens een baanpunt over.
Er zijn nog meer mogelijkheden om tot fraaie plaatjes te komen. We letten weer op de manier van toetsen op de aantrekking van een aantrekkend dekpunt, maar we geven er een andere draai aan. In principe definiëren we een cirkelomgeving van het dekpunt, een kleine cirkel voor een eindig dekpunt en het buitengebied van een heel grote cirkel voor oneindig. We kunnen net zo goed een andere omgeving nemen, een vierkante bijvoorbeeld, of een stervormige enzovoort. Ook kunnen we een cirkelomgeving verdelen in zeg een bovenhelft en een benedenhelft. Een punt waarvan de baan ten slotte voor het eerst terechtkomt in de bovenhelft geven we de kleur rood en anders geven we het een gele kleur. Zo ontstaat op heel eenvoudige wijze een mooi geblokt patroon als omgeving of vulling van een fractal. Die methode is uitgewerkt in het programma JUL1C, slechts een kleine variant van JUL1A. Het programma JUL1B, een andere variant van JUL1A, is gebaseerd op de standaardvorm (11.3). Die versie is vooral geschikt voor situaties waarbij de aard van een dekpunt bijna neutraal is. In het bijzonder zijn mooie plaatjes te verwachten wanneer abs(μ) ongeveer 1 is en wanneer de fase van μ tot π in een eenvoudige rationale verhouding staat (figuur 11.8).



Figuur 11.8 De Julia set van *c* = – 0.8 met binaire codering

Om een Julia set $J(f)$ punt voor punt te construeren zonder te letten op wat er elders gebeurt, kunnen we gebruik maken van het door Julia gevonden resultaat, dat de voorgangers van een enkel punt gelegen op de Julia set met elkaar de Julia set volledig vullen. Het is daartoe wel noodzakelijk om de afbeelding $z' = f(z)$ om te keren, en daartoe moeten we z met een liefst zo eenvoudig mogelijke formule in z' zien uit te drukken. Wanneer die berekening neerkomt op het oplossen van een vierkantsvergelijking, is dat nog wel betrekkelijk eenvoudig te doen. We gaan uit van (11.2) of liever van (11.9). Het is niet moeilijk daaruit het volgende af te leiden:

$$x^2 - y^2 = x_1 - a$$

$$x^2 + y^2 = \sqrt{(x_1 - a)^2 + (y_1 - b)^2}$$

Door optelling en aftrekking, gevolgd door worteltrekking, zijn we in staat de gewenste formules te krijgen die kunnen dienen als basis voor een computerprogramma. Uiteraard krijgen we in het algemeen altijd twee oplossingen, die slechts in teken verschillen. Wat we gedaan hebben kan in complexe notatie heel beknopt geschreven worden als

$$z \rightarrow \pm \sqrt{z - c}$$

maar zolang we computers op de gebruikelijke wijze programmeren in Basic of Pascal zonder subroutines of procedures voor het complexe rekenen, is die mooie complexe notatie alleen nuttig voor theoretische doeleinden. Zoals we al hebben gezegd geeft de omkering van de afbeelding $z \rightarrow f(z)$ aanleiding tot een boomstructuur. Zoals we bij de bespreking daarvan al hadden opgemerkt kunnen we verschillende wegen inslaan. We kunnen op toevallige wijze telkens of de pluswortel, of de minwortel nemen en aldus een grote rij van voorgangers bepalen. Met het programma JULIAMC kan snel voor een willekeurige waarde van c (de twee waarden a en b) een goede benadering van de Julia set worden verkregen. Volgens de zogenaamde Monte Carlo-methode wordt van een willekeurig punt een rij voorgangers berekend. Nadat er een 32-tal punten is overgeslagen kunnen de overige geacht worden op J te liggen.
Deze methode heeft het ernstige bezwaar dat de chaotische wijze waarop de berekende punten zich over J verdelen verre van gelijkmatig is. Sommige gedeelten van J lijken bijna nooit aan de beurt te komen, terwijl andere stukken een overdreven belangstelling krijgen. Dat is allemaal theoretisch wel verklaarbaar, maar in de praktijk hebben we er niet veel aan. Juist de mooiste Julia-fractals blijken de grootste problemen op te leveren. In dergelijke situaties maken we dan liever gebruik van de systematische methode van backtracking, zij het in een aangepaste versie, waarbij we de berekening staken in gebiedjes waar al voldoende punten aanwezig zijn. In het programma JULIABT houden we daartoe een statistiek bij van het aantal punten in zeg 160×120 kleine rechthoekjes waarin we het beeldscherm verdeeld denken. Dat aantal houden we bij in de elementen van een array

van integers. Bij een moderne computer is daarvoor net voldoende geheugenruimte aanwezig. Met dit programma kunnen we gemakkelijk 64 of 128 stappen in de boom afleggen, en dat kan in sommige situaties, waarbij de aantrekking van een dekpunt of cyclus heel zwak is, hard nodig zijn.

Voorbeelden

We vermelden nu ten gerieve van de gebruiker een aantal interessante gevallen. Bij elk voorbeeld geven we de waarden van a,b ($c = a + bi$). Tevens geven we de afmetingen van een geschikt window als halve breedte en halve hoogte. De oorsprong is daarbij steeds het middelpunt. Een window als (2,1) betekent dan dat de Julia set J gelegen is in de rechthoek $-2<x<2$, $-1<y<1$.

1 $a = -1.754878$ $b = 0$ window (2,1)
Er is een superstabiele 3-cyclus.

2 $a = -1.543689$ $b = 0$ window (2,1)
De baan van $z = 0$ bevat een dekpunt. J is een dendriet.

3 $a = -5/4$ $b = 0$ window (1.9,1.1)
Er is een neutrale 2-cyclus $(-1 \pm \sqrt{2})/2$ met $\mu = 1$.

4 $a = -1$ $b = 0$ window (1.8,1.2)
Er is een superstabiele 2-cyclus 0,-1.

5 $a = -3/4$ $b = 0$ window (1.7,1.2)
Het dekpunt $z = -1/2$ is neutraal met $\mu = -1$.

6 $a = 1/4$ $b = 0$ window (1.2,1.4)
Een tweevoudig neutraal dekpunt $z = 1/2$ met $\mu = 1$.

7 $a = 0.3$ $b = 0$ window (1.2,1.4)
Voor $a > 1/4$, $b = 0$ is J altijd een puntenwolk.

8 $a = -0.122561$ $b = 0.744862$ window (1.5,1.4)
Er is een superstabiele 3-cyclus. J staat bekend als 'het konijn van Douady'.

9 $a = -0.101096$ $b = 0.956287$ window (1.6,1.5)
J is een dendriet.

10 $a = 0.11$ $b = 0.67$ window (1.5,1.5)

11 $a = 0.25$ $b = 0.25$ window (1.6,1.5)
Er is een aantrekkend dekpunt.

12 $a = -0.318472$ $b = 0.041257$ window (1.2,1.4)
Er is een superstabiele 11-cyclus.

Julia sets II

De theorie van de Julia sets steunt op de eigenschappen van een conforme afbeelding, een afbeelding waarbij punten in het x,y-vlak volgens een complexe functie $w=f(z)$ worden getransformeerd in punten van een u,v-vlak, waarbij $z=x+yi$ en $w=u+vi$. Een conforme afbeelding heeft als belangrijkste kenmerk dat de grootte van hoeken in het algemeen onaangetast blijft. In het bijzonder wordt een heel klein vierkantje getransformeerd in een ander klein vierkantje, maar wel iets groter of kleiner en wellicht gedraaid. Een conforme afbeelding kunnen we opvatten als een transformatie, als een vertaalregel, waarbij figuren in het ene vlak, het z-vlak, corresponderen met figuren in een ander vlak, het w-vlak. Maar soms is het ook handig het w-vlak te identificeren met het z-vlak, zodat de conforme afbeelding dan een vervorming van figuren in hetzelfde vlak is. Bij de bestudering van Julia sets is die laatste opvatting dan ook van toepassing. Telkens wordt op een punt, een lijn of algemeen op een figuur, dezelfde conforme afbeelding toegepast. Een punt geeft aldus aanleiding tot een baan van punten, een figuur vervormt zich herhaaldelijk, krimpt in of wordt bijvoorbeeld steeds kronkeliger en langer.
In complexe notatie wordt een conforme afbeelding heel beknopt beschreven door

(12.1)
$w=f(z)$

waarbij $f(z)$ een differentieerbare functie is. Voor wie zich alleen met fractals bezighoudt, is $f(z)$ meestal een veelterm in z of een quotiënt van veeltermen. Een enkele maal komt een e-macht, een sinus of een cosinus in aanmerking. Meestal is het noodzakelijk om (12.1) in een reëel en een imaginair deel te splitsen als

(12.2)
$u=u(x,y)$
$v=v(x,y)$

Ter illustratie kiezen we

(12.3)
$f(z)=1/z$

Splitsing in reëel en imaginair deel geeft dan

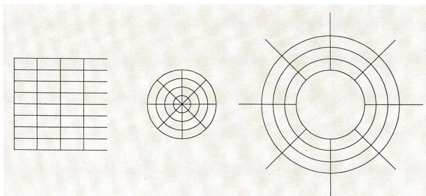
$u(x,y) = x/(x^2+y^2)$
$v(x,y) = -y/(x^2+y^2)$

In dit geval is het handig om in het x,y-vlak poolcoördinaten r,φ in te voeren. Maken we gebruik van de bekende formules
$x = r\cos(\varphi)$, $y = r\sin(\varphi)$, $x^2+y^2 = r^2$, dan volgt hieruit

$u = \cos(\varphi)/r$
$v = -\sin(\varphi)/r$

Wie dit resultaat vergelijkt met de in hoofdstuk 3 besproken meetkundige inversie, zal een grote overeenkomst signaleren. Het enige verschil is eigenlijk het minteken in de tweede regel. De transformatie (12.3) staat bekend als de complexe inversie, en die is identiek met de vermelde meetkundige inversie, mits daaraan nog een extra spiegeling wordt toegevoegd. Immers, de overgang $v \to -v$ is niets anders dan een spiegeling om de reële as. Interpreteren we (12.3) als een transformatie in hetzelfde vlak, dus $z \to 1/z$, dan komt de complexe inversie ook weer neer op een verwisselen van binnengebied en buitengebied van de eenheidscirkel. Kleine cirkels worden grote cirkels en omgekeerd. Middellijnen worden gespiegelde middellijnen. Anders dan bij de meetkundige inversie blijven alleen de punten 1 en -1 op hun plaats.
In figuur 12.1 illustreren we het principe van de conforme afbeelding en tegelijk de complexe inversie. We beginnen met een horizontale strook ruitjespapier $0 < x < \infty$, $0 < y < 2\pi$ (figuur 12.1a). Deze wordt getransformeerd tot het binnengebied van de eenheidscirkel. De transformatiefunctie is niet zo belangrijk, maar kenners zien wel dat het een exponentiële functie is. De horizontale lijnen worden voerstralen van de cirkel en de verticale lijnen worden cirkels. De complexe inversie toegepast op het binnengebied van de eenheidscirkel maakt er het buitengebied van.

Figuur 12.1 Een conforme afbeelding van cirkels en rechte lijnen



Potentiaal en stroomlijn

Het begrip *potentiaal* komt in de natuurwetenschappen veel voor, vooral in verband met de stroming van vloeistoffen of met elektrische ladingen. Bij een zogenaamd potentiaalveld in twee dimensies kunnen we het beste denken aan stroming van vloeistof in grondwater of in oppervlaktewater. Het vloeistofgebied is als het ware verdeeld in een netwerk van potentiaallijnen en stroomlijnen, die elkaar loodrecht snijden. Vloeistofdeeltjes stromen langs de stroomlijnen van punten met een hoge potentiaal naar punten met een lage potentiaal. Ruitjespapier geeft al een eerste voorbeeld. We zien dan als in figuur 12.1 in onze verbeelding vloeistof van links naar rechts lopen langs de horizontale lijnen $y = \text{constant}$. De potentiaallijnen zijn dan de rechten $x = \text{constant}$ met de hoogste potentiaal bij $x = 0$, en de stroomlijnen zijn de horizontale lijnen. In het tweede plaatje zien we hoe dat stromingspatroon in zijn geheel vertaald wordt als een patroon in het binnengebied van een cirkel (figuur 12.1b).
Met een beetje fantasie kunnen we ons voorstellen dat op de rand van de eenheidscirkel op een geheimzinnige manier vloeistof, elektriciteit of iets dergelijks gegenereerd wordt, die vervolgens in het middelpunt van de cirkel verdwijnt.
In het derde plaatje krijgen we een overeenkomstig stromingspatroon in het buitengebied van de cirkel. Op de cirkelomtrek wordt vloeistof gecreëerd die vervolgens langs rechte lijnen als de spaken van een wiel naar het oneindige wegstroomt. In elektrische termen kan de cirkel beschouwd worden als een bliksemafleider, als een geleider waarop zich elektrische lading bevindt. De potentiaallijnen zijn de cirkels en de zich naar het oneindige uitstrekkende lijnen door het middelpunt van de cirkel zijn de stroomlijnen.
Wellicht heeft de lezer zich reeds afgevraagd in hoeverre dit in onze ruimtelijke wereld van toepassing kan zijn. We lijken niets anders te doen dan meetkundige patronen in een tweedimensionale wereld te analyseren. Het antwoord is eigenlijk heel eenvoudig. Wanneer we een driedimensionale situatie hebben waarbij in één lengterichting alles hetzelfde is, dan kunnen we alle ruimtelijke beschouwingen herleiden tot een studie in een plat vlak. Dat is het geval wanneer we bijvoorbeeld iets willen berekenen over een elektrische geleider of een condensator die de vorm heeft van een cilinder. We kunnen ons dan beperken tot het veld van een overeenkomstige cirkelvormige figuur in een vlak loodrecht op de lengte-as van de cilinder.
Ten slotte behandelen we heel summier nog een iets ingewikkelder geval, dat juist voor de door ons bestudeerde Julia-fractals van extra belang is. Het gaat om het elektrische veld van een geleider die, in driedimensionale termen, de vorm heeft van een lange, rechthoekige strook. Uiteraard komt dat hier neer op een vlakke doorsnede waarin zich een lijnvormige geleider bevindt. De wiskundige formulering is uiterst eenvoudig:

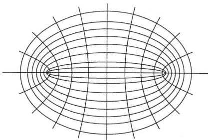
$$f(z) = z + 1/z$$

Het uitgangspunt is het buitengebied van de eenheidscirkel, dus figuur 12.1c. In het w-vlak wordt dat gebied getransformeerd in het buitengebied

van een lijnstuk met de uiteinden $(-2,0)$ en $(2,0)$. Cirkels in het z-vlak met een straal groter dan 1 blijken te worden getransformeerd in ellipsen waarvan de punten $(\pm 2,0)$ de brandpunten zijn. De voerstralen die in het z-vlak van de cirkel naar het oneindige toelopen, gaan over in bogen van hyperbolen die dezelfde brandpunten hebben. Interessant is dat de eenheidscirkel zelf bij deze conforme afbeelding 'platgeslagen' wordt tot een dubbeltellend lijnstuk. Door berekening kan dit alles op vrij eenvoudige wijze bewezen worden, maar dat laten we aan de liefhebber over.

Een Julia set als bliksemafleider

Een bliksemflits ziet eruit als een fractal, en omgekeerd zijn er fractals die ons aan elektrische ontladingen doen denken. In het bijzonder ziet de Julia set van de kwadratische familie met $c=i$ er zodanig uit. Maar het verband van Julia sets met elektrische velden gaat veel dieper. In zekere zin kunnen we elke Julia set opvatten als een bliksemafleider, als een drager van elektrische ladingen, als de bron van een potentiaalveld.
Hierboven troffen we al twee heel speciale voorbeelden aan. Voor de kwadratische afbeelding $z \rightarrow z^2$ is de Julia set de eenheidscirkel $\text{abs}(z)=1$ en voor $z \rightarrow z^2-2$ is de Julia set het lijnstuk $-2<x<2$, $y=0$ en daarvan is het potentiaalveld gegeven in figuur 12.2. Dit waren de twee enige gevallen dat J geen echte fractal is, maar nu beschouwen we een samenhangende kwadratische Julia set. We weten dat $z=\infty$ een superaantrekkend dekpunt is en dat J precies de rand van het aantrekkende gebied is. We hebben in het vorige hoofdstuk gezien hoe mooi we dat gebied konden kleuren door te letten op het vluchtgetal van de naar oneindig wegvluchtende banen. We herhalen nog even het principe. Eerst definiëren we een zogenaamde omgeving van het oneindig verre punt, zeg alles wat verder weg van de oorsprong ligt dan de grens 1000. Voor een startpunt in het aantrekkende gebied van oneindig bepalen we het rangnummer van het baanpunt dat voor het eerst in het aantrekkende gebied van oneindig terechtkomt, en dat is het vluchtgetal. Punten ver weg hebben een laag vluchtgetal, maar punten vlakbij de Julia set hebben een hoog vluchtgetal. Het mooie is dat het vluchtgetal opgevat kan worden als de potentiaal van de elektrisch geladen Julia set. De randen van de kleurige patronen rondom de Julia set kunnen dus opgevat worden als potentiaallijnen.
Er is betrekkelijk weinig wiskunde voor nodig om deze beweringen te staven. Uiteraard kan hier maar een kleine impressie van de achterliggende wiskunde gegeven worden en zal men veel op gezag moeten aannemen. De essentie is dat er een afbeelding geconstrueerd kan worden die het buitengebied van een Julia set transformeert in het buitengebied of het binnengebied van een cirkel.



Figuur 12.2 Het veld van een lijnvormige geleider

Wiskundige achtergrond

In dit gedeelte richten we ons meer tot diegenen die de nodige wiskundekennis in huis hebben. Wie alleen maar kleurige plaatjes wil maken, kan dit stukje zonder bezwaar overslaan. Het komt er in grote lijnen op neer dat het buitengebied van een Julia set (punten z) vertaald kan worden in het buitengebied of het binnengebied van een cirkel (punten w of $1/w$). Die vertaling geschiedt met behulp van een functie H, die gemakkelijk berekend kan worden. Cirkels $\text{abs}(w) = \text{constant}$ worden dan terugvertaald in potentiaallijnen, de randen van de kleurenstroken die in sommige plaatjes de Julia set omgeven. De Julia set zelf wordt aldus vertaald in de eenheidscirkel $\text{abs}(w) = 1$.
Bewezen kan worden dat de afbeeldingsfunctie $z = H(w)$ bepaald is door de betrekking

(12.4)
$$H(w^2) = H^2(w) + c$$

een resultaat dat al in de tijd van Julia goed bekend was en waaraan de naam van Boettcher verbonden is. $H(w)$ kan geschreven worden als een reeks van de vorm

(12.5)
$$H(w) = 1/w + h_1 w + h_2 w^2 + \dots$$

die voor $\text{abs}(w) < 1$ convergeert. Wanneer w zich binnen de eenheidscirkel $\text{abs}(w) < 1$ bevindt, ligt $z = H(w)$ in het buitengebied van de Julia set. De grenswaarde $w = 0$, het centrum van de cirkel, correspondeert met $z = \infty$, het aantrekkende oneindig verre dekpunt. De cirkelomtrek $\text{abs}(w) = 1$ wordt door H vertaald in de Julia set zelf. De coëfficiënten $h_1, h_2, \dots$ kunnen op eenvoudige wijze berekend worden, maar we hebben ze niet nodig.
Er zijn twee bijzondere situaties, te weten $c = 0$ en $c = -2$. Wanneer $c = 0$, is de Boettcher-functie $H(w)$ gelijk aan $1/w$. Dit geeft dus de afbeelding $z = 1/w$. Alles blijkt te kloppen. Het binnengebied van de w-cirkel wordt vertaald in een buitengebied dat tevens het buitengebied van de Julia set is, de eenheidscirkel in het z-vlak. Wanneer $c = -2$, is $H(w) = 1/w + w$, iets waarvan men zich met elementair gereken kan overtuigen. De afbeelding $z = w + 1/w$ stemt overeen met de eerder besprokene (12.3), zij het dat de rollen van z en w verwisseld zijn. Ook nu is alles in orde. In het bijzonder gaan de cirkels binnen de w-cirkel $\text{abs}(w) = \text{constant}$ over in ellipsen om een lijnstuk op de x-as, de Julia set als geïllustreerd in figuur 12.2.
De cirkels $\text{abs}(w) = \text{constant}$ binnen de w-cirkel zijn het prototype van lijnen van een potentiaalveld; de stroomlijnen zijn de halve middellijnen.
De afbeeldingsfunctie $H(w)$ vertaalt dat standaardpatroon in een potentiaalveld buiten de Julia set. We stellen ons tot taak de potentiaal uit te rekenen van een punt z dat zich misschien wel vlakbij de Julia set bevindt, en daartoe moeten we uit de betrekking $z = H(w)$ de waarde van $\text{abs}(w)$ zien af te leiden. Dat probleem kan onverwacht gemakkelijk opgelost wor-

den door gebruik te maken van de baan van z.
De Boettcher-functie kan namelijk dienen om de baanpunten z_n, waarbij $z_0 = z$ het beginpunt is, expliciet te beschrijven als

(12.6)
$z_n = H(w \wedge 2 \wedge n)$

waarbij ^ het symbool van machtsverheffen is. Is dus $z = H(w)$, dan is bijvoorbeeld $z_3 = H(w^8)$. We weten dat z_n, aangetrokken door het superaantrekkende dekpunt $z = \infty$, op den duur heel snel groter wordt en dat dienovereenkomstig w heel klein wordt. Wanneer w zo klein is, kunnen we in (12.5) de bijdragen van de termen met de coëfficiënten h wel vergeten. In dat geval is heel eenvoudig $H(w) = 1/w$ al voldoende nauwkeurig. Het verband tussen z_n en w is dus

(12.7)
$w = z_n \wedge (1/2 \wedge n)$

In de praktijk kunnen we aldus voor elk punt buiten een Julia set snel een waarde voor de potentiaal, en tegelijk voor de fase, vinden. We gaan in een baan net zover totdat voor het eerst bijvoorbeeld $\mathrm{abs}(z_n) > 1000$ en dan berekenen we met gebruikmaking van het laatste baanpunt en het vluchtgetal n de complexe waarde van w. De waarde van abs(w) geeft de potentiaal en arg(w) geeft de fase. Een potentiaallijn abs(w) = constant correspondeert buiten de Julia set met punten, een potentiaallijn, met een zelfde vluchtgetal.
Wie het begrip potentiaal in de natuurkunde kent, weet dat we eigenlijk niet van *de* potentiaal mogen spreken, maar alleen van potentiaalverschillen. Aan een willekeurige potentiaallijn mogen we bij wijze van afspraak een gegeven waarde, 0 bijvoorbeeld, toekennen en alle andere potentiaalwaarden daarop betrekken. Die zelfde willekeur treffen we ook bij de Julia set aan. We zijn immers begonnen met een willekeurige omgeving van ∞ vast te stellen. In het bovenstaande was dat het buitengebied van een cirkel met straal 1000. Natuurlijk was elke andere cirkel, mits groot genoeg, even bruikbaar. De doorwerking betreft alleen een vaste verschuiving bij de vluchtgetallen en een overeenkomstige vaste bijdrage bij alle potentiaalwaarden.
Tot dusverre werd de potentiaal in een punt z steeds gemeten door de overeenkomstige waarde van w, maar dat is wat onhandig. Ligt namelijk z vrij dicht bij J, dan is abs(w) bijna 1, en zouden we veel decimalen nodig hebben voor iets bruikbaars. Het is daarom beter om in plaats van abs(w) de potentiaal te meten door

$^2\log(1/{}^2\log(\mathrm{abs}(w)))$

een – met de computer – gemakkelijk te berekenen waarde die zo geconstrueerd is dat er, afgezien van een vaste verschuiving, precies de vluchtgetallen uitkomen.
De mogelijkheid om betrekkelijk gemakkelijk voor willekeurige punten

buiten de Julia set de potentiaal uit te kunnen rekenen, kan aangewend worden om met de pixelmethode nog mooiere, kleurige plaatjes van Julia sets te kunnen maken. We zijn dan niet langer gebonden aan de door vluchtgetallen bepaalde kleurenbanden, maar we kunnen, voorzover de techniek ons toestaat, bijvoorbeeld langzaam verlopende kleuren gebruiken.

De Julia set op afstand gehouden

Met slechts een kleine uitbreiding van de wiskundige techniek kan nog meer bereikt worden. In het bijzonder zijn we in staat voor de afstand van een punt z tot aan de Julia set, geschreven als $d(z,J)$, een praktisch goed bruikbare benadering af te leiden. Een dergelijke formule is in de praktijk heel waardevol, in het bijzonder in situaties waar de Julia set erg vertakt en 'draderig' is en we de Julia set als enkele lijn, zonder omringende kleuren, op het scherm willen krijgen. In bepaalde gevallen, en vooral wanneer de Julia set juist heel mooi is, blijken gebruikelijke methoden als die van backtracking ons in de steek te laten. Het lijkt wel of dan sommige delen van J heel instabiel zijn of zelden bezocht worden. De bedoelde afstandsformule berust op de eenvoudige overweging dat in het w-vlak de afstand van w tot de cirkelrand gelijk is aan $1 - \mathrm{abs}(w)$. Een infinitesimale afstand in het w-vlak wordt vertaald in een overeenkomstige afstand in het v-vlak volgens de formule $dz = H'(w)dw$. We passen die formule bij wijze van schatting toe op onze situatie, waarbij de afstanden wel klein, maar niet infinitesimaal zijn. Dus

$$d(z,J) = \mathrm{abs}(H'(w))(1 - \mathrm{abs}(w))$$

De afgeleide van $H(w)$ in het rechterlid kunnen we berekenen door gebruik te maken van formule (12.6) of (12.7). In principe zijn we klaar en rest ons slechts wat wiskundige techniek. Het resultaat is ten slotte een heel handige benaderingsformule

(12.8)

$$d(z,J) = T/N$$

waarbij teller en noemer gegeven zijn door

$$T = \mathrm{abs}(z_n)\log(\mathrm{abs}(z_n)) \quad \text{en} \quad N = \mathrm{abs}(w)\mathrm{abs}(dz_n/dz)$$

Het standaardprogramma van de Julia set met de pixelmethode behoeft daarbij slechts een weinig uitgebreid te worden. De enige essentiële aanpassing is dat tegelijk met de baan $z_1, z_2, \ldots$ van een willekeurig punt z ook de afgeleiden dz_1/dz, $dz_2/dz, \ldots$ berekend worden. De baan stopt zodra $\mathrm{abs}(z_n)$ groot genoeg is, en daarna kan de afstandsformule gemakkelijk toegepast worden. In het computerprogramma JULDIST is het een en ander gerealiseerd. Met behulp daarvan kunnen dan fraaie illustraties, zoals figuur 12.3, verkregen worden.

Sommige Julia sets zijn boeiend om hun onverwachte, onregelmatige aspecten, andere bekoren juist om hun meetkundige symmetrie. In dit gedeelte stellen we ons als doel mooie fractals te ontwerpen die een draaiingssymmetrie bezitten als die van zeesterren en Gothische ramen in een middeleeuwse kathedraal. Een wiskundige denkt daarbij aan een complexe veelterm als $z^m - 1$ ($m > 2$), welke de eigenschap bezit dat diens nulpunten op de hoekpunten van een regelmatige m-hoek liggen. We laten het even zien voor het geval $m = 4$. De vier hoekpunten zijn dan 1, i, -1, $-i$ en die vormen met elkaar een vierkant. Anderzijds laat een simpele berekening zien dat elk van die vier getallen, verheven tot de vierde macht, 1 oplevert. Gewapend met deze wetenschap zal het geen verwondering wekken dat een Julia set bepaald door

$$z \rightarrow zP(z)/Q(z)$$

waarbij P en Q veeltermen van z^m zijn, de gewenste symmetrie bezitten. Een wereld van symmetrische fractals gaat nu open. De eenvoudigste keuze is

(12.9)
$$z \rightarrow z(c + (1-c)z^m)$$

waarbij c een willekeurige, complexe constante is. We hebben daarbij al enige schaling toegepast, waardoor er een krans van m dekpunten ontstaat op de eenheidscirkel, $z = 1$ en nog $m - 1$ andere. Daarnaast is ook $z = 0$ een dekpunt en doet ook het oneindig verre punt nog mee als superaantrekkend. De oorsprong heeft de eigenwaarde c en is dus aantrekkend voor $\text{abs}(c) < 1$ en afstotend voor $\text{abs}(c) > 1$. In het laatste geval reikt de Julia set dus tot de oorsprong via m stralen. De eigenwaarde in $z = 1$, en tegelijk van de overige dekpunten op de cirkel, is gelijk aan $1 + m(1-c)$ hetgeen ons weer iets zegt over het te verwachten gedrag van de Julia set. In het programma JULROT1 is het speciale geval $m = 4$ nader uitgewerkt. De volgende keuze is

(12.10)
$$z \rightarrow z(c + (1-c)/z^m)$$

Het lijkt maar een kleine verandering, maar de Julia set is nu totaal anders. De oorsprong is geen dekpunt meer en ∞ is nu een gewoon dekpunt dat aantrekkend of afstotend kan zijn naarmate $\text{abs}(c)$ groter of kleiner is dan 1. In het laatste geval strekt de Julia set zich oneindig ver uit. Maar wel is er een symmetrische serie van dekpunten op de eenheidscirkel, te beginnen met 1. Hun eigenwaarden zijn hier gelijk aan $1 - m(1-c)$. Programmatechnisch is er een klein probleem, omdat het rechterlid voor $z = 0$ oneindig groot kan worden, maar dat kan op verschillende manieren ondervangen worden. Het programma JULROT2 laat zien hoe een en ander werkt (figuur 12.4).

Wie nog meer wil, kan zich uitleven op

(12.11)
$$z \to z(c_1 + c_2 z^m)/(1 + z^m)$$

waarbij twee willekeurige parameters c_1 en c_2 zijn, in totaal dus een keuze van vier (!) reële getallen. In dit geval zijn in het algemeen zowel $z = 0$ als $z = \infty$ dekpunten en is er nog de gebruikelijke serie van m op een cirkel. Het dekpunt $z = 0$ heeft de eigenwaarde c_1, terwijl de aard van $z = \infty$ wordt bepaald door c_2.

De kubische familie

Het ligt voor de hand om na de (te) veel bestudeerde kwadratische Julia sets te vervolgen met door een veelterm van de derde graad bepaalde afbeelding, dus

$$z \to az^3 + bz^2 + cz + d$$

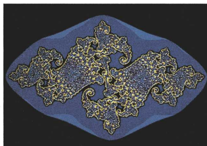
Door schaling kan men dat systeem reduceren tot een soort standaardvorm, waarbij het aantal vrije parameters zo klein mogelijk is. In recente jaren hebben wiskundigen als Branner en Blanchard daaraan veel aandacht besteed, en zij hebben daarbij de volgende eenvoudiger voorstelling gebruikt

(12.12)
$$z \to z^3 - 3a^2 z + b$$

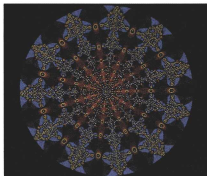
Omdat daarbij zowel a als b als complexe grootheden moeten worden opgevat, betekent dat een viertal vrijheidsgraden wat neerkomt op een enorm aantal mogelijkheden. De keuze van (12.12) is bepaald door een eenvoudige ligging van de kritieke punten. Afgezien van $z = \infty$ zijn deze $z = a$ en $z = -a$. Volgens Julia hangt de aard van de door (12.12) bepaalde Julia set af van de banen van die twee punten. Er zijn kennelijk drie mogelijkheden. Gaan beide banen naar oneindig, dan is J stofachtig als een Cantor-verzameling. Zijn beide banen begrensd, dan is J boogsamenhangend. In overblijvende gevallen is J weliswaar niet samenhangend, maar lijkt als het ware te bestaan uit een verzameling min of meer onderling gelijkvormige 'gewone' Julia sets. Het programma JULCUB stelt ons in staat met het systeem (12.12) werkelijk eindeloos te experimenteren. In figuur 12.5 laten we een enkel fraai exemplaar van de kubische familie zien.

Biomorfie

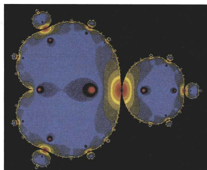
Sommige gevulde of gekleurde fractals doen denken aan biologische objecten, op wat we door een microscoop zouden kunnen zien. Pickover, een Amerikaan die zich veel met computer graphics en met fractals heeft beziggehouden, spreekt van 'biomorphs'. In zijn publikaties toont hij ons daarvan een aantal suggestieve voorbeelden en tevens geeft hij een paar



Figuur 12.3 Een Julia set verkregen met de afstandsformule



Figuur 12.4 Een Julia set met rotatiesymmetrie



Figuur 12.5 Een kubische Julia set

begeleidende formules. In het algemeen gaat het om Julia sets die bepaald zijn door een veelterm met een tweetal hogere machten

(12.13)
$$z \rightarrow c_1 + c_2 z^m + z^n$$

waarbij $m < n$ is en c_1, c_2 willekeurige complexe getallen zijn.
Omdat $z = \infty$ altijd superaantrekkend is, is de Julia set altijd begrensd. Meestal is n vrij groot en dan overheerst de term z^n reeds zodra abs(z) iets groter is dan 1. Dat betekent dat J zich niet veel verder kan uitstrekken dan de eenheidscirkel. Verder is J stofachtig of samenhangend, net als bij de kwadratische familie. Ook is $z = 0$ een kritiek punt, hetgeen betekent dat diens baan in het algemeen naar een aantrekker toegaat, hetzij $z = \infty$, hetzij een dekpunt of een cyclus. Voor de kleuring van de Julia set kunnen we in eerste instantie gebruik maken van de aantrekking van het oneindig verre dekpunt.
Tot dusverre hadden we die kleuring gebaseerd op een cirkelvormige omgeving van ∞, dus het buitengebied van een grote cirkel. Zodra een baanpunt in dat gebied terechtkomt, zodra dus bijvoorbeeld abs(z_n) > 100, interpreteren we het rangnummer n als het vluchtgetal van het beginpunt van de baan, en daaruit leiden we een kleuring af. Om puur wiskundige redenen maakt het niets uit of we een cirkelvormige omgeving of een vierkante omgeving van een aantrekkend punt kiezen, maar om artistieke redenen kunnen de consequenties heel aantrekkelijk zijn. De randen van de gekleurde banden om de Julia set zijn in zekere zin herhaalde kopieën van de rand van wat als omgeving gedefinieerd is. Is die rand een zuivere cirkel, dan zien we van buiten naar binnen gerekend aanvankelijk nog mooie ronde, bijna cirkelvormige banden, maar naarmate we de Julia set naderen, worden die banden meer gegolfd en steeds grilliger. Gaan we daarentegen uit van een vierkante omgeving, zoiets als abs(x) > 100 en abs(y) > 100, dan vertonen zich meteen al bij de buitenste kleurbanen zigzag-vormen en haaietanden. Met de volgende, weinig tijdrovende, toets hebben we zelfs een hele serie van vierkante of stervormige omgevingen:

$\text{abs}(x) + d\,\text{abs}(y) > g$

gecombineerd met

$\text{abs}(y) + d\,\text{abs}(x) > g$

waarin d een getal is tussen 0 en 1 en g een groot getal als 10 of 100. Voor $d = 0$ geeft dit een vierkante omgeving, voor $d = 1/10$ een stervormige omgeving. Daarbij moeten we er wel op letten of we deze twee ongelijkheden in de 'and' of in de 'or' combinatie toepassen. Een zelfde soort toets kunnen we ook toepassen bij een eindig dekpunt, mits bekend. Is de oorsprong $x = 0$, $y = 0$ een aantrekkend dekpunt, dan kunnen we dezelfde formule gebruiken, mits het andere ongelijkteken wordt genomen en g dan een heel klein getal is.
Het programma JULCELL geeft een voorbeeld hoe deze toets gebruikt kan worden. Daarbij is de Julia set bepaald door het speciale geval

$z \rightarrow c + z^3$

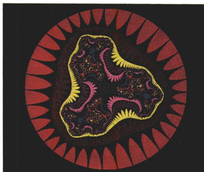
De enkele hogere macht in het rechterlid impliceert weer een rotatiesymmetrie van J als getoond in figuur 12.6.
Het aantal interessante speciale gevallen van (12.13) is praktisch onbegrensd. Hier beperken we ons tot een enkel voorbeeld:

$z \rightarrow c + z^4 + z^8$

Het rechterlid wijst al op een viervoudige rotatiesymmetrie, en inderdaad blijkt de Julia set de vorm te hebben van een vierkant met flauw gebogen zijden, heel geschikt om er een tegeltableau mee te maken als in figuur 12.7.

Sinus, cosinus en e-macht

Het is goed mogelijk om ook met de genoemde functies een fraaie Julia set te ontwerpen, maar daar komt wiskundig wel wat meer voor kijken. In de appendix 1 over de complexe getallen kan men een aantal formules vinden die laten zien hoe $\exp(z)$, $\sin(z)$ en $\cos(z)$ in reëel en imaginair deel gesplitst kunnen worden. We bespreken een tweetal voorbeelden. In de eerste plaats



Figuur 12.6 Een cel van Julia



Figuur 12.7 Tegels van Julia



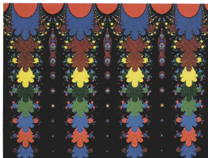
Een detail van de Mandelbrot set na een vergroting van 20 miljard

$z \to \exp(z) - z - 1$

met de in de programma's te gebruiken splitsing

$x \to e^x \cos(y) - x - 1$
$y \to e^x \sin(y) - y$

Kenners zien meteen dat $z = 0$ een dekpunt is dat zelfs superaantrekkend is. Verder is de Julia set periodiek in de y-richting met periode 2π. Bij de afbeelding is het dus al voldoende wanneer we ons tot een horizontale strook van de breedte 2π beperken. De door dat type formules bepaalde Julia sets zijn in recente jaren door onder andere Devaney bestudeerd. Hij heeft laten zien dat onder bepaalde omstandigheden het hele complexe vlak als Julia set kan gelden, weinig interessant voor wie van mooie plaatjes houdt. Net als bij de door veeltermen bepaalde Julia sets moet weer op oneindig getoetst worden, maar dan moeten we alleen letten op x in de positieve richting, dus IF X>XT THEN..., waarbij XT een drempelwaarde is, bijvoorbeeld 10. In figuur 12.8 tonen we de bewuste Julia set voor de waarden verkregen met het programma JULEXP.



Figuur 12.8 Een exponentiële Julia set

Wie zelf met verwante functies, $c + \exp(z)$ bijvoorbeeld, wil experimenteren, dient bedacht te zijn op programmeerproblemen met meldingen als underflow of overflow. Het is daarom raadzaam zo mogelijk met dubbele precisie te werken.

Van de goniometrische familie noemen we de Julia set van de afbeelding

$z \to c \sin z$

waarbij de splitsing in reëel en imaginair berust op

$\sin z = \sin x \cosh y + i \cos x \sinh y$

Ook nu is $z = 0$ een dekpunt, maar het is een gewoon dekpunt met de eigenwaarde c. Weer is de Julia set periodiek, maar nu in verticale stroken van de breedte 2π. In figuur 12.9 geven we een illustratie van een van de mogelijkheden berustend op het programma JULSIN. Wie moed heeft, kan zich wagen aan experimenten met formules als $z \to c + \sin z$. Al met al een interessant en weinig ontgonnen terrein, maar niet zonder technische problemen.



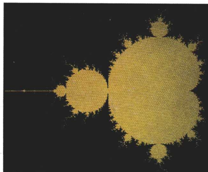
Figuur 12.9 Een Julia set van de goniometrische familie

Mandelbrot sets

Mandelbrot heeft op de hem eigen wijze beschreven hoe hij de nu naar hem genoemde verzameling heeft ontdekt. Maar nu twijfelt men enigszins aan de oorspronkelijkheid van zijn vondst. Het staat inmiddels wel vast dat het probleem waarvan de Mandelbrot set de illustratie is, al veel eerder was gesteld. In 1952 had de Hongaarse wiskundige F. Riesz het bewuste probleem reeds ter oplossing voorgedragen in een lokaal Hongaars tijdschrift. In 1956 verscheen daarvan zowel een Franse als een Russische vertaling, met de voetnoot dat Riesz het probleem gesteld had zonder daarvan de oplossing te kennen. In 1978 lieten de Amerikaanse wiskundigen Brooks en Matelski de waarschijnlijk eerste computerillustratie van de Mandelbrot set zien in een nog wat ruwe vorm. In '79-'80 werd die figuur door Mandelbrot (her)ontdekt en op experimentele wijze beschreven (zie figuur 13.1). Mandelbrot deed veel om er verdere bekendheid aan te geven, en hij inspireerde een aantal wiskundigen zoals Adrien Douady en John Hubbard tot fraai wiskundig onderzoek. Zij waren het die de term 'Mandelbrot set' introduceerden, misschien meer om aan de door hen bestudeerde figuur een soort 'etiket' te kunnen verlenen, gemakkelijk om ernaar te kunnen verwijzen, dan met de bedoeling de 'uitvinder' daarmee te eren. In de natuurwetenschappen worden belangrijke resultaten in de regel toegeschreven aan een bepaalde persoon, maar meestal is dat niet degene die voor het eerst met het goede idee te voorschijn kwam. Wetenschappelijk onderzoek is een gecompliceerd bedrijf, waarbij nieuwe resultaten doorgaans slechts tot stand kunnen komen door gebruik te maken van wat door voorgangers al eerder bedacht is. In het algemeen maken wiskundigen zich niet zo druk over zaken van prioriteit, maar wanneer een lid van de wiskundige gemeenschap wat erg hoog van de toren blaast, kan er wel eens een zodanige commotie ontstaan dat deze via een periodiek als de *Scientific American* bijvoorbeeld tot het algemene publiek doordringt. Wij blijven dus onbekommerd van een Mandelbrot set spreken, trouwens wel met het accent op het onbepaalde lidwoord 'een'. Wat we gewoonlijk onder *de* Mandelbrot set verstaan, is slechts een enkel, maar wel een heel belangrijk exemplaar van een uitgebreide familie van soortgelijke verzamelingen. Uitgangspunt is een iteratief dynamisch systeem van de vorm

$$P \rightarrow f(P, a, b, c, \ldots)$$

waarbij P een punt op een lijn, in het vlak of in de ruimte is, en waarbij a, b, c,... reële parameters zijn. Van belang is de aard van de limietverza-



Figuur 13.1 De Mandelbrot set van $z \to z^2 + c$

meling van het dynamische systeem, en in het bijzonder de vraag voor welke combinaties van parameterwaarden die structuur verandert.

Het bifurcatiediagram van Verhulst

We vormen voor de iteratieve afbeelding $P \to f(P,\ldots)$ een zogenaamd bifurcatiediagram, door de parameters op te vatten als de Cartesische coördinaten van een ruimte en daarin alles af te beelden wat betrekking heeft op de aard van de limietstructuren. Dat bifurcatiediagram duiden we aan als *een* Mandelbrot set, ook al heeft Mandelbrot daar zelf wellicht niet zo diep over nagedacht. Het klinkt juist door zijn algemeenheid misschien wat vaag, maar we zullen later in dit hoofdstuk een paar interessante voorbeelden beschouwen. Het eerste voorbeeld is overigens het bekende bifurcatiediagram van het Verhulst-model:

(13.1)
$x \to ax(1-x)$

dat kort besproken is in hoofdstuk 4.
Daarbij is x een reëel getal, meetkundig een punt op een lijn, en a een reële parameter. De aard van de limietverzameling hangt af van de waarde van a en ook dat getal kunnen we opvatten als een punt van een lijnstuk. De bifurcatiepunten vormen met elkaar een bepaald patroon op de x-as, dat we ook een bifurcatiediagram of bifurcatieset kunnen noemen. Gaande van links naar rechts beginnen we met $a=0$ als eerste bifurcatiepunt. Tot aan $a=1$ blijft de oorsprong $x=0$ aantrekkend, maar wordt voorbij die waarde afstotend. Het punt $a=1$ behoort dus eveneens tot de bifurcatieset. Daarna komt $a=3$, de grens waarop een aantrekkend dekpunt overgaat in een 2-cyclus. Vervolgens gaat de 2-cyclus bij $a=3.4495$, of exact $1+\sqrt{6}$, over in een 4-cyclus, en dat is het begin van de bekende Feigenbaum-reeks waarvan de bifurcatiepunten telkens periodeverdubbeling markeren. We noemen nog even het opvallende venster dat correspondeert met een aantrekkende 3-cyclus. Die verschijnt voor het eerst voor $a=1+\sqrt{22}=3.8284$ en eindigt door bifurcatie tot een aantrekkende 6-cyclus voor $a=3.8415$.
In plaats van het bovenstaande Verhulst-model beschouwen we liever de variant

(13.2)
$z \to z^2 + c$

die er volkomen equivalent mee is. Het verband tussen a en c is

$c = a/2 - a^2/4$

De interessante waarden van a liggen tussen 0 en 4, zodat dan c tussen -2 en $1/4$ in ligt. Het hierboven vermelde rijtje van bifurcatiepunten waar periodeverdubbeling optreedt, is voor c, geordend naar afnemende grootte:

$1/4 \quad -3/4 \quad -5/4 \quad -1.3681 \quad -1.3940 \quad -1.3996 \quad \ldots$

Vooruitlopend op komende dingen kunnen we constateren dat deze punten 'gewoon' de doorsnede vormen van de Mandelbrot set met zijn symmetrieas. De Mandelbrot set is in feite niets anders dan een bifurcatiediagram van het gewijzigde Verhulst-model (13.2) voor *complexe* waarden van c. Juist de uitbreiding van alleen reële waarden van z en c tot complexe waarden geeft aanleiding tot een wereld van Julia sets en een atlas van Mandelbrot. Zoals het reële bifurcatiediagram ons van alles vertelt over de aard van de banen in het Verhulst-model, zo geeft de Mandelbrot set ons alle informatie over de banen in het complexe vlak. De structuur van de banen in het complexe vlak weerspiegelt zich in de aard van de bijbehorende Julia set van (13.2).
Een bifurcatiediagram bepaalt een indeling van een familie dynamische systemen, hier dus de door (13.2) bepaalde kwadratische familie, in klassen. Keren we nog even terug tot het originele model (13.1) waarbij de reële parameter a tussen 1 en 4 in ligt, dan markeert de bifurcatieset een aantal intervallen, waarbij elk interval voor het dynamische systeem een bepaalde karakteristiek geeft. Zo geeft het interval $3 < a < 1 + \sqrt{6}$ aan dat het dynamische systeem een aantrekkende 2-cyclus bezit. Verder correspondeert het interval $-1.7685 < a < -7/4$ met een aantrekkende 3-cyclus. We noemen zo'n interval vaak een *venster* (window). Er zijn aldus oneindig veel van dergelijke vensters waarvoor het systeem telkens een aantrekkende cyclus bezit. Maar toch zijn die vensters niet in staat het interval (1,4) redelijk te overdekken. Er blijven nog (onaftelbaar) oneindig veel a-waarden over waarvoor het systeem zich chaotisch gedraagt.

Het bifurcatiediagram van Mandelbrot

Zoals we zojuist opgemerkt hebben is de Mandelbrot set de uitbreiding van het bifurcatiediagram van Verhulst naar complexe waarden van de parameter c. Traditioneel schrijven we c als $a + bi$, waarbij we als in een x,y-diagram a horizontaal en b verticaal uitzetten. Het bifurcatiediagram is nu een tweedimensionale uitgebreidheid, terwijl dat van het Verhulst-model slechts op een enkele dimensie betrekking had. Men lette er op dat we in beide gevallen met dezelfde kwadratische vorm (13.2) werken. In dat geval is de doorsnede van het Mandelbrot-diagram met de a-as, de lijn waarvoor $b = 0$, identiek met het Verhulst-diagram. De eerste bifurcatiepunten op de a-as zijn $c = 1/4$ en $c = -3/4$. In het Verhulst-model markeren zij de grenzen waarbinnen het dynamische systeem een enkel (eindig) aantrekkend punt bezit. In de complexe situatie is het meteen ingewikkelder. In het algemeen bezit (13.2) twee dekpunten als wortels van de vierkantsvergelijking $z^2 - z = c$. Is z een dekpunt, dan is diens eigenwaarde $2z$. De voorwaarde voor aantrekking is $\text{abs}(2z) < 1$, en deze wetenschap is voldoende om de grens van het gebied te bepalen waar het dynamische sys-

teem een enkel aantrekkend dekpunt heeft. We noemen dat gebied een 1-cel. De grens van die 1-cel kunnen we in parametervorm karakteriseren door

$a = \cos(2\varphi)/4 - \cos(\varphi)/2$
$b = \sin(2\varphi)/4 - \sin(\varphi)/2$

Hierdoor wordt een hartvormige of niervormige kromme bepaald, een zogenaamde cardioïde of hartlijn. De door deze cardioïde begrensde 1-cel vormt het hoofdgebied van het inwendige van de Mandelbrot set. Julia sets voor c-waarden gekozen binnen de 1-cel zien er dus allemaal eender uit, maar het ligt voor de hand dat er leuke dingen te verwachten zijn wanneer c op of vlakbij de rand van de 1-cel ligt. Bij elke waarde van c binnen de 1-cel behoort een aantrekkend dekpunt met een eigenwaarde μ waarvoor $\text{abs}(\mu) < 1$. De bijzondere waarde $\mu = 0$ waarvoor de Julia set het superaantrekkende dekpunt $z = 0$ heeft, correspondeert met $c = 0$. Dat punt noemen we het centrum van de 1-cel. De bijzondere waarde $\mu = 1$ geeft het randpunt $c = 1/4$ van de 1-cel, de zogenaamde wortel van de 1-cel. Het punt $c = -3/4$ waarvoor $\mu = -1$, heet de top van de 1-cel.
Het interval $(-5/4, -3/4)$ van het Verhulst-model waarvoor dus een aantrekkende 2-cyclus bestaat, correspondeert in het complexe vlak met een gebied, een 2-cel, een cirkelschijf bepaald door

$\text{abs}(z+1) < 1/4$

Is μ de eigenwaarde van de 2-cyclus, dan correspondeert de cirkelomtrek met $\text{abs}(\mu) = 1$. De 2-cel heeft het centrum $c = -1$, de waarde waarvoor de Julia set een superstabiele 2-c-cyclus heeft. De waarde $\mu = 1$ geeft de wortel $c = 3/4$, terwijl $c = -5/4$ met eigenwaarde $\mu = -1$ de top geeft. Het is van belang op te merken dat de top van de 1-cel tegelijk de wortel is van de 2-cel. De eigenwaarde verspringt daarmee van -1 naar $+1$.

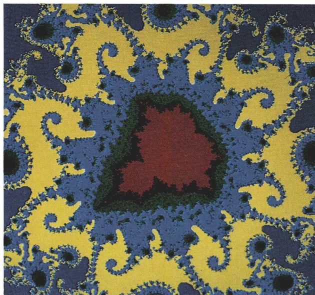
Mandelbrot-miniaturen

Het bekende Feigenbaum-scenario van de herhaalde periodeverdubbeling, oorspronkelijk ontdekt bij het Verhulst-model (13.1) of (13.2) voor reële waarden van c, manifesteert zich in de Mandelbrot set als een reeks van aan elkaar geregen cirkelvormige schijfjes waarvan de diameter afneemt met de Feigenbaum-factor 4.669. De bovenbeschreven 2-cel gaat naar links over in een 4-cel, die op zijn beurt weer tot een 8-cel leidt, enzovoort. Overigens is alleen de 2-cel zuiver cirkelvormig, de andere zijn dit slechts benaderingsgewijs. Telkens is de top van de ene cel weer de wortel van een hogere cel, en telkens verspringt de eigenwaarde van -1 naar $+1$. Ten slotte komen we terecht in een merkwaardig punt $c = -1.4011553$, dat naar Feigenbaum of naar Myrberg genoemd is.
Het antennegedeelte tussen dit punt en de spits $c = -2$ is uiterst merkwaardig, en pas in deze tijd zijn we in staat ook thuis met een enigszins moderne computer die merkwaardigheden onder ogen te krijgen. Overigens is het daarvoor wel nodig om vergrotingen van de orde van zeg een miljoen of

een miljard te gebruiken. De antenne bevat allereerst een aantal miniatuurtjes van het geheel, populair aangeduid als zogenaamde Mandelbroodjes of Mandelbrot-baby's. Dat is weliswaar geen echte zelfgelijkvormigheid, maar veel lijkt het niet te schelen. In de eerste plaats letten we op het bekende 3-venster van het Verhulst-model. In figuur 4.4, waarin het bifurcatiepatroon van het Verhulst-model op een andere wijze is afgebeeld, is dat venster duidelijk zichtbaar. Tevens zien we dat zich ook daar een herhaling van het Feigenbaum-scenario voordoet. De aantrekkende 3-cyclus gaat door bifurcatie over in een aantrekkende 6-cyclus die zich weer splitst in een 12-cyclus enzovoort. In het Mandelbrot-diagram weerspiegelt dit zich in een herhaling van de sequentie 1-cel, 2-cel, 4-cel. Aan het begin staat een 3-cel die 'uit chaos geboren is'. Rechts van de wortel is een stukje antenne, maar naar links gaande is er eerst een klein hartvormig gebied, een 3-cel, dat overgaat in een cirkelvormige 6-cel en zo al verdubbelend steeds verder. Elk celletje is weer gemarkeerd door de centra waarvoor de periodieke cyclus superaantrekkend is, en bij elke overgang zijn er weer grenspunten die tegelijk wortel van het ene en top van het andere gebied zijn.

De antenne op de a-as vanaf -2 naar rechts zit vol met dergelijke miniatuurtjes. Bijna allemaal ontsnappen ze aan onze waarneming, maar bij elke verdere vergroting komen er steeds meer te voorschijn. Wanneer we de omgeving van de Mandelbrot set van kleuren voorzien, zien we dat de miniatuurtjes getrouwe kopieën van het totaal zijn, maar dat de omgeving steeds rijker, hariger en interessanter gaat worden (figuur 13.2).

Figuur 13.2 Een vergroting van een miniatuur

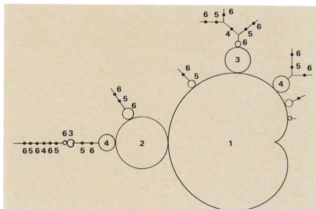


Tot dusverre hebben we nog niet veel meer gedaan dan wat we wisten over de Verhulst-bifurcatie te vertalen in overeenkomstige eigenschappen van de Mandelbrot set. Maar er is veel meer. Eerst keren we terug naar het hoofdgebied, de hartvormige 1-cel. Passeert de eigenwaarde μ van het dekpunt de eenheidscirkel met de fase α, gerekend van $-\pi$ tot π, dan treedt in het algemeen weer bifurcatie op. Heeft α tot π een rationale verhouding, dan betekent dit dat na passage een aantrekkende cyclus ontstaat waarbij de periode bepaald is door de verhouding α/π. Wanneer μ zich op de eenheidscirkel bevindt, ligt c op de omtrek van de 1-cel, de bekende cardioïde. Dat betekent dus dat op bepaalde punten van die hartlijn de 1-cel overgaat in een cel met een hogere periodiciteit. We werken dit uit voor het eenvoudigste geval, de bifurcatie tot een 3-cyclus. De corresponderende fases van μ zijn $\pm 2\pi/3$. Een eenvoudige berekening levert de punten $c=(-1 \pm i3\sqrt{3})/8$. Dat zijn dus wortels van twee 3-cellen, de meest opvallende schijfjes onder en boven de Mandelbrot set. In totaal zijn er dus drie 3-cellen, en het is gemakkelijk na te gaan dat er ook niet meer zijn. Op analoge wijze zijn overal langs de rand van de 1-cel kleine schijfjes vastgehecht waar de verhouding α/π rationaal is. De wortels van de twee 4-cellen zijn aldus $(1 \pm 2i)/4$ en die van de 6-cellen $(3 \pm i\sqrt{3})/8$. Die laatste twee punten zijn overigens de meest linksgelegen punten van de hartlijn (zie figuur 13.3).

Het aantal meercellen neemt snel toe naarmate het rangnummer hoger is. Elke cel kan volgens een of andere bifurcatie tussen twee andere cellen in zitten of kan de vorm hebben van een baby Mandelbrot, als uitgangspunt van een nieuw, zelfstandig leven.

Om een indruk te geven van het grote aantal mogelijkheden geven we onderstaand rijtje:

1 1 3 6 15 27 63 120 252 495 1023



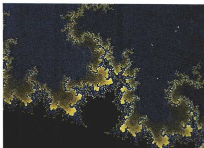
Figuur 13.3 Een schets van de voornaamste bifurcaties

waarbij alleen nog de zogenaamde primitieve cellen worden geteld. Een 3-cel kan natuurlijk ook opgevat worden als bijvoorbeeld een 6-cel of een 9-cel, maar in dit rijtje wordt de cel slechts eenmaal geteld. Er zijn aldus 1023 primitieve 11-cellen.

Theorie

De Mandelbrot set van $z \rightarrow z^2 + c$, kortweg M, wordt gewoonlijk gedefinieerd als de verzameling van die c-waarden waarvoor de baan van $z = 0$, het kritieke punt van de afbeelding, begrensd is. Dat komt in dit speciale geval neer op de baan van c zelf, omdat c na 0 het tweede punt in de baan is. Voor punten buiten M, het complement van M, gaat de baan naar oneindig, de oneindig verre superaantrekker van het systeem. Julia heeft bewezen dat voor c-waarden van M de bijbehorende Julia set samenhangend is, terwijl voor punten buiten M de Julia set stofachtig is. M geeft aldus een globale indeling van Julia sets in twee hoofdtypen, maar we hadden al gezien dat M in feite opgebouwd is uit schijfjes en tussenpuntjes en dat elk onderdeel voor de Julia set iets bijzonders betekent. M is in wezen een bifurcatiediagram waarvan de grens gevormd wordt door bifurcatiepunten waar de Julia set van aard verandert. We kunnen M vergelijken met een atlas waarvan de cellen de landen zijn.

De grenslijn heeft een nauwelijks voorstelbaar ingewikkelde structuur en is een gewild object voor computerexperimenten. In figuur 13.4 geven we er een representatief voorbeeld van. Naarmate we verder vergroten lijkt de rand nog ingewikkelder te worden, en wat we aanvankelijk dachten te zien als haren of antennes blijkt zich op te lossen in een parelsnoer van kleine 'Mandelbroodjes' met nog meer haartjes. De rand is dan ook geen zelfgelijkvormige structuur zoals een gewone fractal, maar blijkt bij elke volgende vergroting iets nieuws te vertonen. Vermoed wordt dat de fractaldimensie zelfs 2 is, een vermoeden dat nog niet is bewezen. Er zijn trouwens nog wel meer onopgeloste problemen, voor wiskundigen even zovele harde noten om te kraken.



Figuur 13.4 Een detail van de rand van de Mandelbrot set

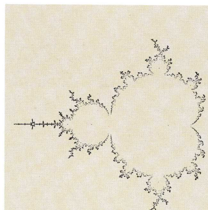
Wiskundige achtergrond

De basis van de wiskundige techniek, nodig om de desbetreffende programma's samen te stellen, wordt gevormd door de baan van het kritieke punt $c = 0$

(13.3)
$$p_0 = 0, \quad p_1 = c, \quad p_2 = c^2 + c, \quad p_3 = c^4 + 2c^3 + c^2 + c, \quad \ldots$$

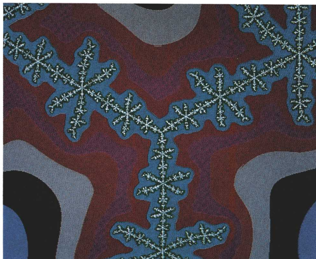
Elk element is een veelterm in c, een veelterm waarvan de graad snel toeneemt als een macht van 2. Voor bijvoorbeeld de index 11 is de graad reeds 1024. Een veelterm heeft in het complexe vlak zoveel nulpunten als de graad bedraagt. De veelterm met index 11 heeft dus 1024 van dergelijke nulpunten. Daartoe behoort in elk geval $c=0$, maar voor de overige moet de computer hard rekenen. Is c een dergelijk getal, dan levert de rij $0, p_1, p_2, \ldots, p_{10}$ een superstabiele 11-cyclus en c is het centrum van een 11-cel waarvoor de Julia set een aantrekkende 11-cyclus bezit. In totaal zijn er een eerder vermeld aantal van 1024 van deze celletjes overal verspreid over M. We bekijken ook nog even het polynoom p_6. Daarbij zijn 32 nulpunten, maar dat zijn niet even zovele 6-cycli. De 0-cyclus voor $c=0$ valt al meteen af. Maar ook de 2-cyclus behorende bij $c=-1$ mogen we niet meetellen, en evenmin de drie 3-cycli welke uit $p_3=0$ volgen. Maar toch blijven er nog 27 echte 6-cycli over. Er zijn dus 27 6-cellen, of schijfjes die door bifurcatie uit grotere schijven, de 1-cel, de 2-cel, en de 3-cellen ontstaan zijn, of de begincellen van Mandelbroodjes. In figuur 13.5 is een groot aantal nulpunten van de p-veeltermen afgebeeld. Met elkaar vormen ze het bekende Mandelbrot-patroon.
Naast de centra, de wortels en de toppen kunnen we bij de Mandelbrot set nog zogenaamde Misiurewicz-punten onderscheiden, de beeldpunten van c-waarden waarvoor de Julia set een dendriet is. Tegelijk zijn dat de waarden waarvoor de baan van het kritieke punt $z=0$ een periodieke cyclus bevat zonder dat $z=0$ daartoe behoort. Treedt die cyclus bij het rangnummer k voor het eerst op en is m de periode, dan is dus voldaan aan de gelijkheid

(13.4)
$$p_k = p_{k+m}$$


Figuur 13.5 Het Mandelbrot-patroon gevormd door centra

Uiteraard wordt in de p-rij de cyclus daarna telkens herhaald. Het eenvoudigste geval waarbij de periode 1 is, is $p_2=p_3$ en daaruit volgt $c=-2$. Inderdaad is de Julia set in dit geval een lijnstuk, maar dat was al bekend. Het eenvoudigste geval van een 2-cyclus is $p_2=p_4$, en daaruit volgt $c=i$ of $c=-i$. De baan van $z=0$ wordt dan $0, i, -1+i, -i, -1+i, -i, \ldots$ Bewezen is dat de bij een Misiurewicz behorende cyclus altijd instabiel is, zodat de baan van $z=0$ in zijn geheel deel uitmaakt van de Julia set. De rand van de Mandelbrot set is dicht bezaaid met dergelijke bijzondere punten. Een interessant gegeven, want Misiurewicz-punten kenmerken zich door een heel bijzondere ligging. Het zijn of knooppunten van de haren, of uiteinden daarvan. De zojuist genoemde punten $c=-2$ en $c=\pm i$ zijn de uiteinden van de voornaamste antennes van M. In figuur 13.6 is een detail van M in de omgeving van het Misiurewicz-punt $c=-0.1011+0.9563i$

Figuur 13.6 De omgeving van een Misiurewicz-punt



afgebeeld, waarbij met een speciale techniek de haren extra geaccentueerd zijn. De Julia set van een Misiurewicz-punt is een dendriet, en het zal de experimentator opvallen dat die Julia set erg lijkt op de harige omgeving van dat Misiurewicz-punt op de rand van M, wanneer men die tenminste voldoende vergroot. Dat fenomeen kan wiskundig gefundeerd worden, en men kan de rand van M op deze wijze opvatten als een staalkaart van Julia sets, of nog beter als een atlas van Julia-kaarten.

Mandelbrot elektrisch geladen

Net als bij een kwadratische Julia set kan M opgevat worden als een elektrisch geladen geleider, welke een zich tot in het oneindige uitstrekkend veld verwekt met potentiaallijnen en faselijnen. De wiskundige achtergrond berust op de reeds in het vorige hoofdstuk besproken functie $H(w,c)$ (zie formule 12.4). In de notatie hebben we nu de parameter c ook vermeld, om duidelijk te maken dat zowel w als c veranderlijken zijn. Weer doorloopt w de schijf begrensd door de eenheidscirkel, terwijl c zich in het buitengebied van M bevindt. Door de betrekking

(13.5)
$H(w,c)=0$

wordt een verband gelegd tussen die twee variabelen, hetgeen betekent dat het gebied $\text{abs}(w)<1$ op het buitengebied van M wordt afgebeeld. De cirkels $\text{abs}(w)=\text{constant}$ worden dus de potentiaallijnen om de Mandelbrot set. Het is mogelijk om voor c een reeksontwikkeling in machten van w op te stellen:

$c = 1/w - 1/2 + w^2/8 - w^3/4 + \ldots$

maar helaas convergeert de reeks steeds moeizamer naarmate abs(w) dichter bij 1 is. Het is niet eens zo moeilijk om de reeks nog met enige tientallen termen voort te zetten, maar veel helpt het niet. Het kan echter veel beter. Om de afbeelding voor computerberekening geschikt te maken, vervangen we de bovenstaande betrekking door de equivalente vorm

$H(w \wedge 2 \wedge m) = p_m(c)$

waarbij het rangnummer zo groot is, dat we voor het linkerlid een eenvoudige benadering kunnen vinden. Dit geeft het goed bruikbare resultaat

(13.6)
$\log(1/w) = 2^{-m} \log(p_m(c))$

In de praktijk is het voldoende dat we met de baan van 0 net zolang doorgaan totdat bijvoorbeeld abs(p_m) > 1000. De zojuist gegeven formule levert ons dan de complexe potentiaal w voor de gebruikte waarde van c. Zijn we alleen geïnteresseerd in potentiaallijnen, dan dienen we in (13.6) alleen het reële deel te nemen, zodat in overeenstemming met de corresponderende formule voor de Julia set (12.7)

(13.7)
$\text{abs}(w) = \text{abs}(p_m(c)) \wedge (1/2^m)$

Net als bij de Julia sets kunnen we de daardoor berekende waarde in verband brengen met het vluchtgetal. De mooiste toepassing van de theorie van de potentiaal is echter de afleiding van een afstandsformule d(c,M), welke ons op een eenvoudig berekenbare wijze vertelt hoever een punt buiten M ongeveer verwijderd is van de rand. Die formule is

$\text{d}(c,M) = \text{abs}(\text{d}c/\text{d}w)(1 - \text{abs}(w))$

waarbij dc/dw de afgeleide is van de afbeeldingsfunctie $c(w)$. Die afgeleide kunnen we op dezelfde manier berekenen als bij de Julia sets door bij de bepaling van de kritieke baan $p_m(c)$ ook de afgeleiden naar c mee te nemen. Ten slotte komt er weer in overeenstemming met formule (12.8) van de Julia sets

(13.8)
$\text{d}(c,M) = T/N$

waarbij teller en noemer gegeven zijn door

$T = \text{abs}(p_m)\log(\text{abs}(p_m))$ en $N = \text{abs}(w)\text{abs}(\text{d}p_m/\text{d}c)$

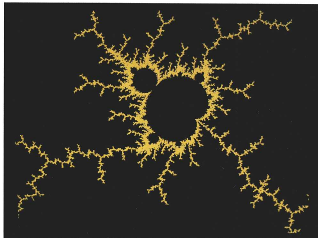
Computerillustraties

Met de tegenwoordige computers is het een vrij eenvoudige zaak ook thuis snel een fraaie afbeelding op het beeldscherm te krijgen. In principe gebruiken we de pixelmethode, zoals bij de Julia set in bijvoorbeeld het programma JUL1A. Eerst selecteren we een rechthoek van het af te beelden gedeelte van M en identificeren we die rechthoek met het beeldscherm, of desnoods een deel daarvan. Elke pixel van de gekozen rechthoek correspondeert dan met een c-waarde waarvoor de baan $p_1, p_2, p_3, \ldots$ getoetst moet worden. De index n waarvoor het baanpunt p_n verder weg ligt dan een van tevoren afgesproken grens, het zogenaamde vluchtgetal, bepaalt de kleurindex. Hoewel we soms zestien kleuren ter beschikking hebben, is het om artistieke redenen in het algemeen raadzaam zich bij de keuze van aantal en aard grote beperkingen op te leggen. Willen we bijvoorbeeld acht kleuren gebruiken, dan kunnen we de index n modulo 8 reduceren zodat de kleurbanden zich om de acht herhalen. Willen we sterk vergrote details op of bij de rand van M afbeelden, dan dienen we bedacht te zijn op hoge waarden van het vluchtgetal, soms in de orde van 1000 of meer. Het is dan nodig de berekeningen in dubbele precisie uit te voeren, iets wat alleen binnen redelijke tijd uitvoerbaar is wanneer de computer uitgerust is met een mathematische coprocessor. Toch zal men ook dan bedacht moeten zijn op berekeningen van vele uren, nachten of dagen.
Het eenvoudigste programma, MANTOT, geeft vrij snel een totaalbeeld van de Mandelbrot set. De variant MANDET geeft op geriefelijke wijze een illustratie van een willekeurig zelf te kiezen detail. Bij de zojuist besproken methode blijken soms de mooiste details van de Mandelbrot set aan de aandacht te ontsnappen. De meestal ragdunne haren lopen doorgaans tussen de pixels door. Punten daarvan komen derhalve slechts weinig aan de beurt. Maar wanneer we gebruik maken van de afstandsfunctie, kunnen werkelijk spectaculaire resultaten worden bereikt. In het programma MANDIST is die methode nader uitgewerkt. Men ziet daarin beter dan met formules dat de berekening van de afstand volgens formule (13.8) slechts weinig extra moeite en tijd kost. In principe moet voor elk punt van het venster waarin we een detail van de rand van M nader willen onderzoeken, die afstand berekend worden. We kunnen dan twee wegen inslaan. We kunnen de grootte van de geschatte afstand vertalen in een kleurwaarde en aldus een detail van de rand omgeven met kleurige banden. Maar ook kunnen we alleen de rand zichtbaar maken door slechts die c-punten te plotten waarvoor de afstandsformule een voldoende kleine waarde bezit, een halve pixelafstand bijvoorbeeld. In de praktijk werken we meestal met een wat flexibele drempel, zodat we de uiterst dunne, harige tentakels nog enigszins kunnen verdikken. In figuur 13.7 tonen we een met het bewuste programma verkregen resultaat, een baby Mandelbrot met veel haar.

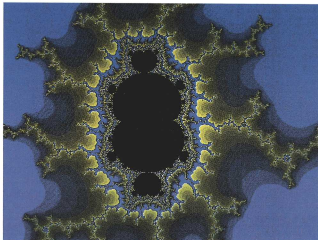
Andere Mandelbrot sets

Er zijn legio mogelijkheden om varianten van de traditionele Mandelbrot set te bedenken. In het algemeen ziet het totaalbeeld van wat we de M set zullen noemen er soms heel anders uit, maar de structuur van de samen-

stellende cellen past in het vertrouwde beeld. Ongetwijfeld tot veler verrassing zijn er vaak miniatuurtjes aanwezig die bij vergroting toch weer sterk lijken op de ons vertrouwde Mandelbrot set van de kwadratische afbeelding. Er zijn daarvoor goede wiskundige redenen, en in zekere zin is de standaard M set werkelijk een wiskundige standaard. Toch is er voor een experimentator veel te beleven, vooral wanneer de afbeeldingsfunctie $f(z,c)$ een symmetrie bevat. Men zij er steeds op bedacht dat de aard van de Mandelbrot set bepaald is door de banen van alle kritieke punten.



Figuur 13.7 Een harig detail van de Mandelbrot set



Figuur 13.8 Een detail van de kubische Mandelbrot set

Voor Mandelbrot sets behorende bij veeltermen van z is het een goede regel om te eisen dat alle kritieke banen naar oneindig gaan.
Van de talloze mogelijkheden laten we er een paar de revue passeren. We beginnen met de M set van

$$z \rightarrow c - z^3$$

welke zowel horizontaal als verticaal symmetrisch is. Die symmetrie hangt samen met het feit dat z alleen in oneven machten voorkomt. In de computerpraktijk betekent dat het prettige gevolg dat maar een vierde deel van het totaal berekend behoeft te worden. De afbeelding heeft $z = 0$ als enig kritiek punt, zodat we als bij de kwadratische M set kunnen toetsen op de baan van $z = 0$. De eigenschappen van de kubische M set zijn als die van de bekende Mandelbrot-figuur. In plaats van het appelmannetje zien we nu een tweevoudig symmetrisch, kussenvormig patroon met knobbels, insnoeringen en uitstulpingen. Ook nu zijn er talloze harige parelsnoeren van kleine miniatuurtjes. In figuur 13.8 laten we zo'n miniatuurtje met tentakels zien.
Algemener zijn er bij de afbeelding

$$z \rightarrow z^3 - 3a^2z + b$$

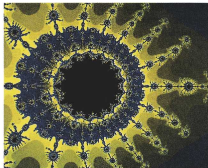
twee complexe parameters a en b om mee te werken. De Mandelbrot set is in dit geval een vierdimensionaal object dat door wiskundigen reeds enigermate is onderzocht. Wanneer we daarin of a, of b constant veronderstellen, krijgen we daarvan tweedimensionale doorsneden die duidelijk verwant zijn met de klassieke Mandelbrot set. Een probleem bij deze afbeelding is dat er twee kritieke punten $\pm a$ zijn, en dat we in principe zowel de baan van $z = a$ als die van $z = -a$ moeten toetsen. Experimenten laten we over aan de lezer.
Wie nog iets anders wil ondernemen, kan het beste een afbeelding van het type

$$z \rightarrow c + z^2P(z)$$

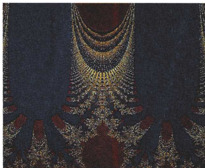
kiezen waarbij $P(z)$ een eenvoudige veelterm met getallencoëfficiënten is. In dat geval is $z = 0$ altijd een kritiek punt en kunnen we beginnen met alleen de daardoor bepaalde baan te toetsen. De andere kritieke punten volgen uit $P'(z) = 0$. Wil men ook hun banen in de beschouwing betrekken, dan moeten we bij de keuze van P ervoor zorgen dat die punten bekende eenvoudige getallen zijn. In figuur 13.9 tonen we tot slot van deze serie een mooi detail van de M set van $c + z^{16}$, waarin zich een miniatuur van de basisfiguur bevindt.

Exponentiële en goniometrische Mandelbrot sets

Heel mooi zijn de Mandelbrot sets van functies waarin leden van de familie exp, sin en cos voorkomen. Om er goed mee te kunnen werken is wat meer wiskunde nodig, en dan nog kunnen experts verrassingen beleven,

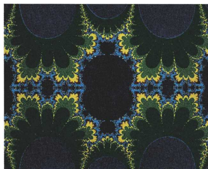


Figuur 13.9 Een detail van een hogere-machts Mandelbrot set



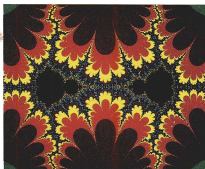
man exp

Figuur 13.10 Een exponentiële Mandelbrot set



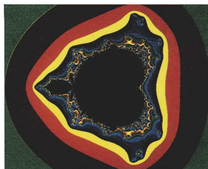
mansin1

Figuur 13.11 Een Mandelbrot set van een sinusfunctie

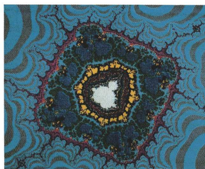


mansin2

Figuur 13.12 Een andere goniometrische Mandelbrot set



Figuur 13.13 Een variant van de bekende Mandelbrot set



Een kiemcel van Mandelbrot (vergrotingsfactor 3×10^9)

soms van theoretische aard, soms van technische aard. In het algemeen dienen de berekeningen in dubbele precisie uitgevoerd te worden, maar dan nog kunnen meldingen als overflow en underflow optreden. Ook kunnen problemen ontstaan doordat functies als sin x niet goed gedefinieerd zijn wanneer x erg groot is. Evenals bij de overeenkomstige Julia sets geven we enkele voorbeelden. In figuur 13.10 geven we de Mandelbrot set van

$$z \to c + z - \exp(z)$$

waarbij het programma MANEXP gebruikt is.
Het rechterlid is zodanig gekozen, dat $z = 0$ een kritiek punt is en we dus op de baan van dat punt kunnen toetsen. De dekpunten volgen uit $\exp(z) = c$. Ze zijn aantrekkend voor $\text{abs}(c - 1) < 1$, zodat we de daardoor bepaalde cirkel al bij voorbaat kunnen uitsluiten. De exponentiële functie is periodiek in verticale richting, en dat zelfde geldt hier voor de M set. Het is voldoende ons bij de afbeelding tot een horizontale strook van de breedte 2π te beperken.
Op overeenkomstige wijze kunnen we een Mandelbrot set maken van de afbeelding

$$z \to c + z - \sin z$$

Wanneer z erg klein is, is het rechterlid bij benadering gelijk aan $c + z^3/6$, wat wil zeggen dat de M set verwantschap vertoont met die van $c + z^3$. Ook hier is het voldoende te toetsen op de baan van $z = 0$ als kritiek punt. De periodiciteit van de sinus en de vorm van de gekozen afbeelding maken dat de M set opgebouwd is uit zich periodiek herhalende, verticale stroken die elk de breedte 2π hebben. In figuur 13.11 geven we een representatieve illustratie waarvoor het programma MANSIN1 gebruikt is.
Er kunnen ook fraaie Mandelbrot sets gemaakt worden van functies als $c_1 + c_2 \sin(z)$, waarbij er twee willekeurige constanten zijn. Wie ermee wil werken, dient in principe te toetsen of de banen van alle kritieke punten in de y-richting naar oneindig gaan. In de regel zijn hierbij, afgezien van de altijd aanwezige periodiciteit, slechts de twee kritieke punten $\pm\pi/2$ van belang. In figuur 13.12 laten we nog een aardig speciaal geval zien, de Mandelbrot set van $c + \pi \sin(z)$ op grond van het programma MANSIN2, een symmetrisch geval waarbij de twee kritieke banen equivalent zijn.

Een pseudo-Mandelbrot set

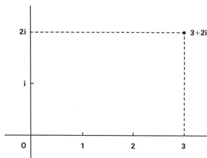
Op het Mandelbrot-thema zijn eindeloos veel variaties mogelijk, maar ook hier moeten we ons tot een enkel voorbeeld beperken. Het ligt een beetje voor de hand de klassieke Mandelbrot set als uitgangspunt te nemen en daar een kleine verandering in aan te brengen. Een van de mogelijkheden is om niet te toetsen op een kritieke baan, maar op de baan van een willekeurig punt Q. Is P het beeldpunt van het complexe getal c in de kwadratische afbeelding $z \to z^2 + c$, dan kunnen we algemeen gesproken zowel P als Q variëren. Bij de klassieke Mandelbrot set was Q de oorsprong als kri-

tiek punt en werd de kleuring uitgevoerd voor het vluchtgetal van de baan van dat punt naar oneindig. Maar er is eigenlijk niets op tegen om in plaats van de oorsprong een willekeurig punt Q te nemen. In het computerprogramma MANVAR is dit slechts een heel kleine verandering. Totaal tellen P en Q voor vier vrijheidsgraden, zodat we feitelijk te maken hebben met een vierdimensionale uitgebreidheid. De traditionele Mandelbrot set is hiervan dan een vlakke doorsnede. In figuur 13.13 laten we zien wat er gebeurt als in plaats van de oorsprong een punt op de y-as wordt genomen. Zoals we zien heeft het bekende appelmannetje vreemde sprieten gekregen en is ook de samenhang enigszins verloren gegaan. Overigens ligt ook hier weer een heel terrein braak voor uren computerplezier.

Het complexe vlak

Het is hier niet de plaats een uitvoerige beschouwing te houden over complexe getallen en over complexe analytische functies. Maar wel zullen we trachten in kort bestek enkele essentiële punten samen te vatten in de hoop dat dan ook de niet-deskundige lezer een iets beter begrip krijgt van de theoretische achtergrond van de mooie plaatjes die hij als computerkunst wellicht reeds onder ogen heeft gehad. In het bijzonder dient deze appendix ter ondersteuning van de meer wiskundig georiënteerde hoofdstukken over de Julia- en de Mandelbrot-verzamelingen.
Het valt wellicht op dat meetkundige en rekenkundige begrippen vaak door elkaar worden gebruikt, alsof punt en getal identiek zijn. Zoiets komt ook al voor in de natuurkunde bij de theorie van het licht. Nu eens praten we over golven en golfpakketjes, en dan weer praten we over fotonen alsof we met kleine kogeltjes worden gebombardeerd. In de wiskunde hebben we primair te maken met een x,y-coördinatenstelsel. Men kan denken aan getallenparen (a,b) en deze (getallen)vectoren noemen, maar men kan ook denken aan een netwerk van elkaar loodrecht snijdende lijnen en dan (a,b) opvatten als de coördinaten van een snijpunt. Tegenover de getallenvectoren staan dan meetkundige vectoren als pijltjes. In de wiskunde, en met name in beschouwingen omtrent complexe getallen, zijn beide zienswijzen onverbrekelijk verbonden.

Bij een meetkundig punt met de coördinaten (a,b) denken we tegelijk aan een zogenaamd complex getal dat we symbolisch schrijven als $a + bi$ of als $a + ib$, waarbij het symbool i de zogenaamde imaginaire eenheid is. De positie van i doet er niet toe, dat is meer een kwestie van smaak of van duidelijkheid in de typografie. De gewone getallen a identificeren we met complexe getallen waarvoor $b = 0$ is. Als punten beschouwd liggen ze op de x-as, de vertrouwde getallenlijn.
Complexe getallen c zijn aldus meetkundig gedefinieerd als punten in het platte vlak of rekenkundig als tweetallen gewone getallen. Schrijven we $c = a + bi$, dan noemen we a het reële deel en b het imaginaire deel. We gebruiken daarbij vaak de notatie (zie figuur 1)



Figuur 1 Complex getal en beeldpunt

$a = \mathrm{Re}(c)$
$b = \mathrm{Im}(c)$

Het complexe getal $a - bi$ heet de *geconjugeerde* van $a + bi$. Meetkundig corresponderen twee geconjugeerde getallen met punten die ten opzichte van de x-as een gespiegelde positie innemen.
Eigenlijk berust de hele complexe rekenkunde op de afspraak dat

$$i^2 = -1$$

Verder rekenen we op de gebruikelijke wijze zoals we dat op school geleerd hebben. We laten hieronder een paar nuttige voorbeelden volgen die men zelf gemakkelijk kan narekenen.
Uit $c = a + bi$ en $z = x + yi$ volgt dan

$$cz = (ax - by) + (ay + bx)i$$

en verder

$$z^2 = (x^2 - y^2) + 2xyi$$

en

$$z^3 = x(x^2 - 3y^2) + y(3x^2 - y^2)i$$

Bij de kwadratische Julia sets hebben we te maken met de kwadratische afbeelding

$$z' = z^2 + c$$

Die formule kunnen we zien als een compacte schrijfwijze, waarin twee betrekkingen zijn samengevat. Nemen we van linker- en rechterlid eerst de reële delen apart en vervolgens de imaginaire delen, dan krijgen we

$$x' = x^2 - y^2 + a$$
$$y' = 2xy + b$$

Dat ziet er wat ingewikkelder uit, maar het is wel de vorm die we in programma's moeten gebruiken. Voor theoretische beschouwingen is de complexe formulering veel beter.

Absolute waarde en argument

Bij complexe getallen z werken we veel met de begrippen *absolute waarde* $\mathrm{abs}(z)$ en *argument* $\arg(z)$ (of fase). In formulevorm

$$\mathrm{abs}(z) = \sqrt{x^2 + y^2} \quad \text{en} \quad \arg(z) = \arctan(y/x)$$

Bij het argument doet zich een complicatie voor die in computerprogramma's tot problemen kan leiden. Het argument wordt namelijk gerekend van $-\pi$ tot π met een sprong bij de negatief reële as ($y=0$, $x<0$) terwijl de arctangens maar van $-\pi/2$ tot $\pi/2$, en verder periodiek, geteld wordt.
In meetkundige zin heeft de absolute waarde van z de betekenis van de afstand van het punt $P(z)$ tot de oorsprong $O(0)$, dus $\text{abs}(z)=OP$. Het argument is dan gelijk aan de hoek die de vector OP met de positieve x-as maakt. Zijn z_1 en z_2 complexe getallen die corresponderen met de punten P_1 en P_2, dan kan de lengte P_1P_2 dus berekend worden als $\text{abs}(z_1-z_2)$.
Soms is het prettig om voor een complex getal zowel met reëel en imaginair deel als met absolute waarde en argument te rekenen. Dat is eigenlijk hetzelfde als het tegelijk gebruiken van Cartesische x,y-coördinaten en poolcoördinaten.
Is r de absolute waarde van het complexe getal $z=x+yi$ en φ het argument, dan geldt het bekende verband tussen Cartesische coördinaten en poolcoördinaten

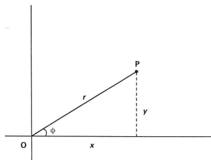
$$x=r\cos(\varphi) \quad y=r\sin(\varphi)$$

zodat z geschreven kan worden als (figuur 2)

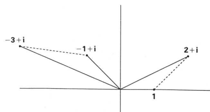
$$z=r(\cos(\varphi)+i\sin(\varphi))$$

Absolute waarde en argument kunnen vooral nuttig zijn bij vermenigvuldigen, machtsverheffen en worteltrekken. De vermenigvuldiging van twee complexe getallen komt namelijk neer op het vermenigvuldigen van de absolute waarden en het optellen van de argumenten (figuur 3).
Voor machtsverheffen geldt dienovereenkomstig de regel

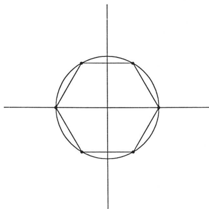
$$z^m=r^m(\cos(m\varphi)+i\sin(m\varphi))$$



Figuur 2 Absolute waarde en argument



Figuur 3 Vermenigvuldigen van complexe getallen



Figuur 4 Complexe getallen als hoekpunten van een regelmatige zeshoek

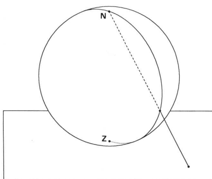
Complexe getallen waarvan de absolute waarde 1 is, corresponderen met punten op de eenheidscirkel $r = 1$. Bij Julia sets hebben ze vaak de betekenis van bifurcatiepunten, waarbij aantrekking overgaat in afstoting. Van belang voor bijvoorbeeld Julia-fractals met een rotatiesymmetrie is de wetenschap dat de punten bepaald door

$$z = \cos(k\varphi) + i\sin(k\varphi)$$

waarbij $\varphi = 2\pi/m$ en k het rijtje $0, 1, 2, \ldots, m-1$ doorloopt, op de hoekpunten van een regelmatige m-hoek liggen. Een eenvoudige berekening laat zien dat altijd voldaan is aan $z^m = 1$. We constateren dat aan die vergelijking voldaan is door m verschillende oplossingen, een speciaal geval van de zogenaamde hoofdstelling van de algebra die zegt dat een willekeurige algebraïsche vergelijking van de graad m in het algemeen m oplossingen heeft. Juist het bestaan van die eigenschap voor complexe getallen maakt dat de klassieke Mandelbrot set zo rijk aan details is (figuur 4).

De bol van Riemann

In de wereld van de complexe getallen is er een enkel oneindig groot getal ∞, meetkundig het oneindig verre punt van het vlak. Eigenlijk wringt dat een beetje, omdat we in een vlak het oneindige in verschillende richtingen kunnen benaderen. Men zou op meetkundige gronden eerder een oneindig verre lijn verwachten. Het is daarom beter het meetkundige beeld van een plat vlak met complexe getallen ietwat aan te passen en liever te denken aan een heel grote bol die in de oorsprong aan het complexe vlak raakt. De zuidpool van de bol correspondeert in die visie met het getal 0 en de noordpool met het getal ∞. Die simpele gedachte heeft veel voordelen. We kunnen ons daarmee bijvoorbeeld een betere visuele voorstelling maken van hoe een Julia set zich naar het oneindige uitstrekt. De bol als alternatieve wereld van complexe getallen wordt gewoonlijk Riemann-bol genoemd, naar Bernhard Riemann (1826-1866), een grondlegger van de theorie van de complexe functies (zie figuur 5).



Figuur 5 De bol van Riemann

Bij Julia en Mandelbrot sets komen we telkens het begrip omgeving van een punt of complex getal tegen. In beginsel is dat een klein cirkeltje of een klein vierkantje om het bewuste punt. Correspondeert dat met het complexe getal $c = a + bi$, dan kan een cirkelomgeving eenvoudig worden beschreven als $\text{abs}(z - c) < \delta$ waarin δ een klein positief getal is. In computerprogramma's dienen we dit te vertalen als

$$(x - a) \times (x - a) + (y - b) \times (y - b) < \text{eps}$$

waarbij eps de veelgebruikte aanduiding is voor een klein getal.
Voor een vierkante omgeving geldt analoog

$$\text{abs}(x - a) < \text{eps} \quad \text{en} \quad \text{abs}(y - b) < \text{eps}$$

Het begrip omgeving kan ook voor het oneindig verre punt ∞ gebruikt worden. Een cirkelomgeving is niets anders dan het buitengebied van een grote cirkel, dus $\text{abs}(z) > R$, waarbij R een groot getal is. In computerprogramma's schrijven we dan

$$x \times x + y \times y > R \times R$$

Delen en worteltrekken

Delen door een getal is hetzelfde als vermenigvuldigen met het omgekeerde. De omgekeerde van bijvoorbeeld z kunnen we weer in de standaardvorm van een complex getal brengen door gebruik te maken van de eenvoudig te verifiëren gelijkheid

$$(x + yi)(x - yi) = x^2 + y^2$$

Hieruit volgt dus

$$1/z = x/(x^2 + y^2) - iy/(x^2 + y^2)$$

Om uit een complex getal c de vierkantswortel z te kunnen trekken is meer nodig. In de eerste plaats zijn er twee vierkantswortels, die onderling alleen in teken verschillen. Worteltrekken komt neer op het oplossen van de vergelijking $z^2 = c$. Wanneer we zowel voor z als voor c poolcoördinaten gebruiken, gaat het nog vrij eenvoudig als volgt

$$\pm\sqrt{c} = \pm\sqrt{r}\,(\cos(\varphi/2) + i\sin(\varphi/2))$$

waarbij r en φ absolute waarde en argument van c zijn.
Om van een als $c = a + bi$ gedefinieerd complex getal de vierkantswortels te kunnen berekenen, moeten we op deze wijze eerst de absolute waarde en het argument bepalen, wat niet altijd even prettig is. Met de volgende in programma's veel toegepaste techniek, in JULIABT bijvoorbeeld, kan het ook. We stellen $w = u + vi$ en $z = x + yi$ waarbij $w^2 = z$, zodat $w = \pm\sqrt{z}$.
Uitgeschreven betekent dit

$u^2 - v^2 = x$ en $2uv = y$

Met wat schoolse wiskunde kunnen we hieruit afleiden dat

$$u^2 + v^2 = \sqrt{x^2 + y^2}$$

Van x^2 en y^2 kennen we nu zowel som als verschil, zodat ze door optellen en aftrekken snel gevonden kunnen worden. Ten slotte is

$$u = \pm\sqrt{x + \sqrt{(x^2 + y^2)/2}}$$

en

$$v = \pm\sqrt{-x + \sqrt{(x^2 + y^2)/2}}$$

Nu moeten we even goed opletten om geen programmeerfouten te maken. De tekens van de rechterleden dienen zodanig gecombineerd te worden dat voldaan is aan $2uv = y$. Is bijvoorbeeld y positief, dan moeten in beide regels dezelfde tekens gekozen worden. In het programma betekent dat dus een extra regel.

Complexe transformaties

Sommige meetkundige transformaties kunnen met de complexe schrijfwijze veel eenvoudiger beschreven worden. We laten het overigens bij een enkel representatief voorbeeld.
De complexe vermenigvuldiging

$$z' = \mu z$$

beschrijft bijvoorbeeld een gelijkvormigheidstransformatie waarbij de oorsprong het centrum is. De lineaire schaalfactor is abs(μ) en de rotatiehoek is arg(z) (zie ook figuur 3). Wanneer abs(μ) < 1, is de transformatie een contractie, een krimpdraaiing. Alle punten die bij een transformatie op hun plaats blijven, heten dekpunten (fixed points). Hier is de oorsprong $z = 0$ een dekpunt, maar in het complexe vlak doet ook $z = \infty$ mee als tweede dekpunt. Vatten we $z \rightarrow \mu z$ op als een dynamisch systeem, dan wordt de transformatie dus eindeloos herhaald. Is abs(μ) < 1, dan gaan alle banen naar $z = 0$ toe, zodat dat dekpunt aantrekkend is. Met het beeld van de Riemann-bol voor ogen kijken we nu naar $z = \infty$. Onder een beginpunt in een omgeving van ∞ denken we nu aan een punt dat zich binnen een cirkelschijfje rond de noordpool bevindt. Diens baan loopt dan weg van de noordpool en spiraliseert naar beneden in de richting van de aantrekkende zuidpool. De noordpool is een afstotend dekpunt en de zuidpool is een aantrekkend dekpunt. Is abs(μ) > 1, dan hebben we een situatie die er eigenlijk op neerkomt dat de rol van de polen is verwisseld. De zuidpool, het dekpunt $z = 0$, stoot af en de noordpool, het dekpunt $z = \infty$, trekt aan. Is abs(μ) $= 1$, dan beschrijft $z' = \mu z$ een zuivere draaiing over de hoek arg(μ). De Riemann-bol roteert dan om de as die de polen met elkaar verbindt.

De transformatie

$$z' = 1/z$$

is identiek met de in hoofdstuk 3 beschreven cirkelinversie, mits daar nog een spiegeling om de x-as aan toegevoegd wordt. Is P het beeldpunt van z en P' dat van z', dan is dus $OP' = 1/OP$.
Dit alleen al betekent dat punten binnen de eenheidscirkel en punten buiten de eenheidscirkel verwisseld worden. De punten 1 en -1 blijven bij de transformatie op hun plaats en zijn dus de dekpunten. Verder kan vastgesteld worden dat het argument van z' de tegengestelde is van die van z. Voor de Riemann-bol betekent figuur 4 dat de bol als geheel geroteerd wordt om een horizontale as die door de punten 1 en -1 van de equator gaat. Bij de draaiing, een halve slag, verwisselen noordpool en zuidpool van plaats. Wanneer de transformatie $z' = 1/z$ tweemaal achtereen wordt uitgevoerd, staat alles weer op zijn plaats, net als bij een spiegeling. Een dynamisch systeem kan er dus niet van gemaakt worden.

Differentiëren

Een enkele maal is het nodig een complexe functie te differentiëren, bijvoorbeeld wanneer we de eigenwaarde willen berekenen van een dekpunt of van een periodieke cyclus van een dynamisch systeem als een Julia set. De regels van het differentiëren zijn die van gewone functies. Overigens beperken we ons hoofdzakelijk tot polynomen. Uit de definitie van differentiaalquotiënt, de zogenaamde afgeleide, volgt

$$f(z) - f(z_0) \approx f'(z_0)(z - z_0)$$

wanneer z dicht bij z_0 ligt. Dat kunnen we op de volgende wijze interpreteren. De transformatie $z' = f(z)$ is in het klein gelijkwaardig met een gelijkvormigheidstransformatie waarvan de complexe schaalfactor gelijk is aan de plaatselijke afgeleide. Wanneer overigens $f'(z_0) = 0$, dan geldt dit niet meer. We hebben dan te maken met een zogenaamd kritiek punt.
Een goed voorbeeld is de functie z^2 in de omgeving van $z = 0$. We besluiten dit gedeelte met het voorbeeld

$$f(z) = cz - (1 - c)z^2$$

waarvoor de Julia set in het programma JUL1B berekend wordt. Er zijn de twee dekpunten 0 en 1. Berekening geeft

$$f'(z) = c - 2(1 - c)z$$

zodat $f'(0) = c$ en $f'(1) = 2 - c$. De transformatie $z' = f(z)$ is in de omgeving van het dekpunt dus gelijkwaardig met de gelijkvormigheidstransformatie $z' = cz$, een krimpdraaiing voor abs$(c) < 1$. Bij de bespreking van de Julia sets hadden we gezegd dat c de eigenwaarde van het dekpunt 0 is, en dat dat dekpunt aantrekkend was voor abs$(c) < 1$. Analoog heeft het dekpunt

1 de eigenwaarde $2-c$ en is dit aantrekkend voor $\mathrm{abs}(2-c)<1$. Lokaal volgt dat weer uit de benadering $z'-1 \approx (2-c)(z-1)$.

De exponentiële functie

We betreden nu het terrein waar wiskundigen zich nog plezierig voelen, maar anderen moeten afhaken. We moeten daarom hier heel kort zijn en we vermelden slechts enkele formules als tovermiddelen voor adembenemende Julia en Mandelbrot sets. De exponentiële functie $\exp(z)$ wordt voor $z=x+yi$ gedefinieerd als

$$\exp(z)=\exp(x)(\cos(y)+i\sin(y))$$

In alle opzichten gedraagt de complexe exponentiële functie zich als zijn bekende reële versie $\exp(x)$. Alle gewone rekenregels gaan ongewijzigd door, ook het differentiëren. Verder is de functie periodiek in de imaginaire richting, dus

$$\exp(z+2\pi i)=\exp(z)$$

In de x-richting is er een heel steil verval van heel klein tot heel groot. Voor $x\rightarrow\infty$ schiet $\mathrm{abs}(\exp(z))$ komeetachtig naar oneindig en voor $x\rightarrow-\infty$ duikt $\mathrm{abs}(\exp(z))$ op nul af. Wie met de exponentiële functie op de computer werkt, moet daarom wel bedacht zijn op foutmeldingen als overflow of underflow. De periodiciteit van de exponentiële functie komt weer te voorschijn bij de corresponderende Julia en Mandelbrot sets die ook een periodieke opbouw in stroken laten zien.
Uit de bovenstaande definitie volgt dat de goniometrische functies nauw verweven zijn met de exponentiële functie. Curiositeitshalve vermelden we de zogenaamde formules van de Moivre

$$\cos(z)=(\exp(zi)+\exp(-zi))/2$$
$$i\sin(z)=(\exp(zi)-\exp(-zi))/2$$

Van groter belang voor de praktijk is de splitsing van de complexe cosinus en sinus in reëel en imaginair deel

$$\cos(x+yi)=\cos(x)\cosh(y)-i\sin(x)\sinh(y)$$
$$\sin(x+yi)=\sin(x)\cosh(y)+i\cos(x)\sinh(y)$$

waarbij de zogenaamde hyperbolische cosinus cosh en de hyperbolische sinus sinh gedefinieerd zijn als

$$\cosh(y)=(e^y+e^{-y})/2$$
$$\sinh(y)=(e^y+e^{-y})/2$$

Computertips

Eerst een paar opmerkingen over gebruikte notaties. Bij de programma's gebruiken we altijd hoofdletters. Uiteraard moeten wiskundige formules altijd op één regel geplaatst kunnen worden. In de tekst passen we ons daarbij zoveel mogelijk aan, zij het dat we voor indices en exponenten wel gebruik maken van subscript en superscript. Om typografische redenen werken we een enkele keer met het in Basic gebruikte symbool ^ voor de machtsverheffing. Zo is 2 ^ 3 gelijk aan 8 en is 2 ^ 2 ^ 3 gelijk aan 2 ^ 8 of 256. Het log-symbool wordt alleen gebruikt als de natuurlijke logaritme, die het getal e = 2.71828... als basis heeft.
De programma's zijn geschreven in Basic, waarbij alleen die opdrachten gebruikt zijn welke in praktisch alle versies voorkomen. Ze kunnen aldus ongewijzigd gebruikt worden als broncode voor gecompileerde versies verkregen met bijvoorbeeld Turbo Basic, Power Basic (de opvolger van Turbo Basic), of met Quick Basic. Beperkt men zich in het gebruik tot een dergelijke versie, dan kunnen sommige programma's wel in een beter gestructureerde vorm gebracht worden, bijvoorbeeld bij lussen waarin op meer kenmerken getoetst wordt.
Sommige programma's kunnen dienen als voorbeeld wanneer men voor verwante problemen zelf programma's wil opstellen. Bij het edit-proces kan men veel nut hebben van ingebouwde stops of pauzes.
Voor een pauze is de opdracht

```
A$ = INPUT$(1)
```

heel geschikt. De computer wacht dan totdat men een willekeurige toets indrukt. Een andere mogelijkheid is bijvoorbeeld

```
IF INKEY$<>"" THEN END
```

Zodra een willekeurige toets ingedrukt wordt, stopt de computer automatisch. Een dergelijke faciliteit is heel plezierig wanneer men met tijdrovende programma's in bijvoorbeeld Turbo Basic aan het experimenteren is.
Een grote zorg is het altijd te kleine werkgeheugen. Soms is het prettig met grote arrays te kunnen werken, bijvoorbeeld met de gegevens van alle pixels in een rechthoekig deel van het beeldscherm. Een vol beeldscherm van 640×400 pixels is zelden haalbaar, een vierde deel wil nog wel eens, maar meestal moet men bescheidener zijn. Om het onderste uit de kan te krijgen kan men proberen het type van de arrays zo eenvoudig mogelijk te

houden. Arrays van integers nemen minder geheugenruimte in, en soms kan men zelfs gebruik maken van speciale typen als short integers. Ook kan men telkens informatie heen en weer spelen tussen werkgeheugen en schijfgeheugen, maar dat laatste is vaak tamelijk ingewikkeld. Wanneer men een plaatje wil vormen met een horizontale of een verticale symmetrie behoeft men eigenlijk alleen de helft, of een kwart, van het totaal te berekenen. Voor elk berekend punt kunnen dan op het beeldscherm twee of vier gekleurde pixels geplaatst worden.
De volgende zorg is de grote tijdsduur van vaak de meest interessante pixelprogramma's. De berekening van een detail van de Mandelbrot set kan best enige dagen in beslag nemen, tenminste wanneer we een beeldschermvullende illustratie willen. In het begin kan men daarom beter experimenteren met een plaatje in een klein gedeelte van het scherm, iets als een grote postzegel. Ook kan men aanvankelijk een concessie doen aan de nauwkeurigheid. Het is in het algemeen verstandig de definitieve programma's zodanig te bewerken, dat op geregelde tijden de informatie opgeborgen wordt, zodat een berekening op willekeurige tijdstippen onderbroken kan worden om later, op de stille uren, voortgezet te worden.

Kleuren

Met kleuren kan heel veel gedaan worden. Wanneer we daarbij esthetische argumenten laten prevaleren, is het raadzaam spaarzaam met kleuren om te springen. In principe zijn er zestien standaardkleuren beschikbaar, en kan men die kleuren ook nog selecteren uit een totaal van 64. Bij bijvoorbeeld Quick Basic kan men de kleuren opbouwen uit een rode, een groene en een blauwe component in verschillende intensiteiten. Daarbij kan een uit zestien willekeurige kleuren bestaand palet worden gevormd, bijvoorbeeld als een geleidelijke reeks van geel naar blauw.
Bij de programma's in dit boek is geen gebruik gemaakt van de PALETTE-faciliteit. In het algemeen beperken we ons tot de zestien standaardkleuren, zodat bijvoorbeeld PSET(a,b),1 neerkomt op het plaatsen van een blauwe stip. Om in hetzelfde programma toch de mogelijkheid te hebben van een andere kleurenkeuze, zijn de zestien standaardkleuren verdeeld over een array als COL(.) gedefinieerd met behulp van een DATA-lijst. Een opdracht als PSET(a,b),COL(1) plaatst dan een stip in de door COL(1) bepaalde kleur. Wanneer de PALETTE-opdracht gebruikt kan worden dient men onderscheid te maken tussen de begrippen kleurattribuut en kleur. PSET(a,b),1 betekent in feite dat aan de pixel op plaats (a,b) het kleurattribuut 1 toegekend is. Wordt nu een ander kleurenpalet gekozen, dan betekent het attribuut 1 gewoon de eerste kleur van het gekozen palet. Met een handig gebruik van de PALETTE-opdracht kunnen reeds gemaakte kleurenplaatjes in een enkele slag op geheel andere wijze gekleurd worden. Vooral bij Julia sets en Mandelbrot-details wordt deze werkwijze sterk aanbevolen.
In de regel wordt met de pixelmethode voor elke in aanmerking komende pixel een natuurlijk getal k, het vluchtgetal bij de kwadratische Julia sets, bepaald. Is k niet te groot en willen we een palet van bijvoorbeeld acht kleuren gebruiken, dan kunnen we k tot een kleurenattribuut herleiden

via 1 + k mod 8. Het attribuut 0 komt hierbij niet aan de beurt, maar meestal reserveren we dat toch voor zwart. Kan k groot worden, in de orde van 1000, dan kunnen we op allerlei manieren groepjes van k-waarden tot een enkele kleur bij elkaar nemen, bijvoorbeeld $1 + \text{int}(k/10) \bmod 8$. Wie na vele uren computerarbeid een mooi plaatje op een diskette zou willen opslaan, kan misschien zijn voordeel doen met de volgende minder bekende programmaregels. Daarbij maken we gebruik van een paar heel speciale input/output-adressen in een 80 × 86 processor:

```
DEF SEG = &HA000: OUT &H3CE,4
OUT &H3CF,0: BSAVE M$ + ".BLU",0,38400
OUT &H3CF,1: BSAVE M$ + ".GRN",0,38400
OUT &H3CF,2: BSAVE M$ + ".RED",0,38400
OUT &H3CF,3:BSAVE M$ + ".INT",0,38400
OUT &H3CF,0
```

Het komt erop neer dat het plaatje dat zich aan het eind van een programma op het beeldscherm bevindt met deze extra regels op een gewenste diskette opgeborgen wordt als een groepje van vier bestanden. Hebben we bijvoorbeeld een Mandelbrot-detail gemaakt en willen we het plaatje naar de diskette in B sturen, dan moeten we M$ vervangen door iets als B: MANDEL1. De tweede opdracht brengt het blauwe gedeelte van het plaatje naar de file B: MANDEL1.BLU. Met de derde en de vierde opdracht worden evenzo het groene en het rode gedeelte opgeborgen. Ten slotte wordt het intensiteitsgedeelte naar een vierde file weggeschreven. De laatste opdracht herstelt de situatie van de desbetreffende output ports. Op een high density-schijfje van 3.5″ is plaats voor negen plaatjes, totaal dus 36 files, en voor nog een paar aanvullende programma's. Wanneer we van de opgeborgen files later weer plaatjes op het beeldscherm willen maken, kan dat heel eenvoudig met een leesprogramma dat aan het begin het volgende groepje opdrachten bevat

```
DEF SEG = &HA000: OUT &H3C4,2
OUT &H3C5,1: BLOAD M$ + ".BLU"
OUT &H3C5,2: BLOAD M$ + ".GRN"
OUT &H3C5,4: BLOAD M$ + ".RED"
OUT &H3C5,8: BLOAD M$ + ".INT""
OUT &H3C5,15
```

Men ziet dan de gevraagde illustratie met een soort vierkleurendruk op het scherm verschijnen. Uiteraard is het handig in het leesprogramma een gedeelte op te nemen met behulp waarvan de kleuren naar wens veranderd kunnen worden. Met de PALETTE-opdracht gaat dat heel plezierig.

Computerprogramma's

De volgende 72 computerprogramma's staan per hoofdstuk geordend. Tevens zijn deze fractals verkrijgbaar op 3½ diskette (720 kB). De flop bevat ook de langere programma's welke wel worden genoemd in het boek, maar hieronder niet afgedrukt. Daarnaast zijn er nog een paar extra programma's op de diskette opgenomen. Aanbevolen wordt om de laatste met een basic compiler te combineren. Daartoe is op de schijf ook het MS-DOS besturingsprogramma van Turbo Basic opgenomen. Desgewenst kan men ook een tweetal 5¼ diskettes bestellen. De schijfjes zijn verkrijgbaar door overmaking van f 49,95 op gironummer 55 670 69 van uitgeverij Aramith te Bloemendaal onder vermelding van 'superfractal 3½' of 'superfractal 5¼'.

Hoofdstuk 1

MEANDER
BROWNL
PYT3BT
HENATT

Hoofdstuk 2

SIERLINE
SIERMC
MENGERMC
MENGERCL
LEVYLINE
LEVYLONG
DRAGON
MULTIFR

Hoofdstuk 3

FERN1
FERN2
SQUARE1
SQUARE2
INVERMC

Hoofdstuk 4

LORENZ1
LORENZ2
ROESSLER
VERHULST
BIFUR1
BIFUR2

Hoofdstuk 5

TREESB
TREEBT
TREEBTA
SIERBT

Hoofdstuk 6

TENNIS
CONTRMC1
MAPLE
COLLAGEH

Hoofdstuk 7

RWALK
CLUSTERC
CLUSTERL

Hoofdstuk 8

KOCHLIND
SIERLIND
PEANOL
BUSHLIND

Hoofdstuk 9

PICK1
PICK2
PICK3
CELL1
CELL2
CELL1A
CELL3
CELLC

Hoofdstuk 10	Hoofdstuk 11	Hoofdstuk 12	Hoofdstuk 13
PYT3DBT	JULFILL	JULDIST	MANTOT
PROFILE	JUL1A	JULROT1	MANDET
PROJS	JUL1B	JULROT2	MANDIST
WAERDEN	JUL1C	JULCELL	MANEXP
TAKAGI	JULIAMC	JULCUB	MANSIN1
FRSURF	JULIABT	JULEXP	MANSIN2
		JULSIN	MANVAR

Hoofdstuk 1

```
10 REM ***THE FRACTAL OF MANDELBROT AND GIVEN***
20 REM ***NAME:MEANDER***
30 REM ***FRACTAL LINE ON (0,0),(1,0) AS BASE LINE***
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS : PI=3.141593
70 WINDOW (-.3,-.6)-(1.3,.6)
80 P=4 : DIM T(P) : REM ***ORDER***
90 REM ***THE MOTIF IS COMPOSED OF Q UNIT VECTORS
   WITH A GIVEN***
100 REM ***STARTING POINT AND A GIVEN DIRECTION
    IN DEGREES***
110 Q=6 : DIM A(Q),B(Q),F(Q),C(Q),S(Q) : REM ***
    NUMBER VECTORS OF MOTIF***
120 DATA 0,0,0,.333,.333,-90,.333,.333,0,.667,0,90,
    .667,0,0,.667,0,180
130 FOR I=0 TO Q-1 : READ A(I),B(I),F(I)
140 F(I)=F(I)*PI/180 : C(I)=COS(F(I)) :
    S(I)=SIN(F(I)) : NEXT I
150 FOR N=0 TO Q^P-1 : REM ***MAIN LOOP***
160 M=N : FOR J=0 TO P-1
170 T(J)=M MOD Q : M=M\Q : NEXT J
180 X=0 : Y=0 : U=1 : V=0
190 FOR J=0 TO P-1 : C=C(T(J))/3 : S=S(T(J))/3
200 IF J MOD 2=1 THEN GOSUB 230 ELSE GOSUB 250
210 NEXT J : LINE (X,Y)-(U,V)
220 NEXT N : A$=INPUT$(1) : END
230 Z=X : X=C*X-S*Y+A(T(J)) : Y=S*Z+C*Y+B(T(J))
240 W=U : U=C*U-S*V+A(T(J)) : V=S*W+C*V+B(T(J)) :
    RETURN
250 Z=X : X=C*X+S*Y+A(T(J)) : Y=-S*Z+C*Y-B(T(J))
260 W=U : U=C*U+S*V+A(T(J)) : V=-S*W+C*V-B(T(J)) :
    RETURN
270 A$=INPUT$(1) : END
```

```
10 REM ***BROWNIAN LINE***
20 REM ***NAME:BROWNL***
30 SCREEN 9 : CLS : RANDOMIZE TIMER
40 WINDOW (-1.2,-.9)-(1.2,.9)
50 W=40 : Y=0
60 LINE (1,0)-(-1,0)
70 FOR K=0 TO 2000
80 X=-1+K/1000 : Y=Y+W*(RND-.5)/2000
90 LINE -(X,Y)
100 NEXT K : A$=INPUT$(1) : END
```

```
10 REM ***BARE TREE, BACKTRACKING***
20 REM ***ORDER RESTRICTED BY DIAMETER EPS OF
   SMALLEST SQUARE***
30 REM ***NAME:PYT3BT***
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS : PI=3.141593
70 WINDOW (-2,-2)-(4.4,2.8)
80 EPS=.04 : P=24 : DIM XA(P),YA(P),XB(P),YB(P),S(P)
90 F1=PI/3 : F2=-PI/6 : REM ***PRESCRIBED ANGLES***
100 R1=.55 : R2=.85 : REM ***PRESCRIBED REDUCTIONS***
110 A1=R1*COS(F1) : B1=R1*SIN(F1)
120 A2=R2*COS(F2) : B2=R2*SIN(F2)
130 LINE (0,-1)-(0,0)
140 G=0 : M=0 : XA=0 : YA=-1 : XB=0 : YB=0
150 S=SQR((XB-XA)*(XB-XA)+(YB-YA)*(YB-YA))
160 IF G<P-1 AND S>EPS THEN GOSUB 260 ELSE GOSUB 310
170 IF M<0 THEN A$=INPUT$(1) : END ELSE GOTO 150
180 U=XB-XA : V=YB-YA
190 XX=XB : YY=YB
200 U1=A1*U-B1*V : V1=B1*U+A1*V
210 X1=XB+U1 : Y1=YB+V1
220 U2=A2*U-B2*V : V2=B2*U+A2*V
230 X2=XB+U2 : Y2=YB+V2
240 RETURN
250 LINE (X1,Y1)-(XB,YB) : LINE -(X2,Y2) : RETURN
260 M=M+1 : G=G+1 : GOSUB 180 : GOSUB 250
270 XA=XX : YA=YY : XB=X1 : YB=Y1
280 XA(M)=XX : YA(M)=YY
290 XB(M)=X2 : YB(M)=Y2 : S(M)=G
300 RETURN
310 G=G+1 : GOSUB 180 : GOSUB 250
320 XA=XA(M) : YA=YA(M) : XB=XB(M) : YB=YB(M)
330 G=S(M) : M=M-1 : RETURN
340 END
```

```
10 REM ***HENON ATTRACTOR***
20 REM ***NAME:HENATT***
30 CLS : KEY OFF : PRINT"PRESS 9 FOR EGA OR 12 FOR
   VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 DEFDBL A,B,C,D,X,Y,Z
70 A=1.4 : B=.3
80 R1=-1.5 : R2=-.45 : R3=1.5 : R4=.45 : GOSUB 390
90 X=.6 : Y=.6 : K=1 : REM ***START***
100 WHILE K<10000 AND LEN(INKEY$)=0
110 Z=X : X=1+Y-A*X*X : Y=B*Z
120 IF K>32 THEN PSET (X,Y)
130 K=K+1 : WEND
140 R1=.54 : R2=.15 : R3=.73 : R4=.21 : GOSUB 410
150 A$=INPUT$(1)
160 CLS : GOSUB 390
170 X=.6 : Y=.6 : K=1 : REM ***START***
180 WHILE K<100000! AND LEN(INKEY$)=0
190 Z=X : X=1+Y-A*X*X : Y=B*Z
200 IF K>32 AND ABS(X-XM)<DELX AND ABS(Y-YM)<DELY
    THEN PSET (X,Y)
210 K=K+1 : WEND
220 R1=.621 : R2=.185 : R3=.641 : R4=.191 : GOSUB 410
230 A$=INPUT$(1)
240 CLS : GOSUB 390
250 X=.6 : Y=.6 : K=1 : REM ***START***
260 WHILE K<1000000! AND LEN(INKEY$)=0
270 Z=X : X=1+Y-A*X*X : Y=B*Z
280 IF K>32 AND ABS(X-XM)<DELX AND ABS(Y-YM)<DELY
    THEN PSET (X,Y)
290 K=K+1 : WEND
300 R1=.6304 : R2=.1889 : R3=.6324 : R4=.1895 :
    GOSUB 410
310 A$=INPUT$(1)
320 CLS : GOSUB 390
330 X=.6 : Y=.6 : K=1 : REM ***START***
340 WHILE K<1E+07 AND LEN(INKEY$)=0
350 Z=X : X=1+Y-A*X*X : Y=B*Z
360 IF K>32 AND ABS(X-XM)<DELX AND ABS(Y-YM)<DELY
    THEN PSET (X,Y)
370 K=K+1 : WEND
380 A$=INPUT$(1) : END
390 WINDOW (R1,R2)-(R3,R4) : XM=(R1+R3)/2 :
    YM=(R2+R4)/2
400 DELX=(R3-R1)/2 : DELY=(R4-R2)/2
410 LINE (R1,R2)-(R3,R2) : LINE -(R3,R4) :
    LINE -(R1,R4) : LINE -(R1,R2)
420 RETURN : END
```

```
10 REM ***SIERPINSKI SIEVE***
20 REM ***NAME:SIERLINE***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS : PI=3.141593
60 WINDOW (-2.6,-2.4)-(2.6,1.5)
70 P=5 : DIM T(P) : A=SQR(3)
80 FOR M=0 TO P
90 FOR N=0 TO 3^M-1
100 N1=N : FOR L=0 TO M-1
110 T(L)=N1 MOD 3 : N1=N1\3 : NEXT L
120 X=0 : Y=0
130 FOR K=0 TO M-1
140 X=X+COS((4*T(K)+1)*PI/6)/2^K
150 Y=Y+SIN((4*T(K)+1)*PI/6)/2^K
160 NEXT K
170 U1=X+A/2^(M+1) : U2=X-A/2^(M+1) :
    V1=Y-1/2^(M+1) : V2=Y+1/2^M
180 LINE (U1,V1)-(X,V2)
190 LINE -(U2,V1) : LINE -(U1,V1)
200 NEXT N : NEXT M
210 BEEP : A$=INPUT$(1) : END
```

```
10 REM ***FRACTAL DETERMINED BY THREE CONTRACTIONS***
20 REM ***SPECIAL CASE OF THE SIERPINSKI SIEVE AS A
   CLOUD OF POINTS***
30 REM ***NAME:SIERMC***
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS : RANDOMIZE 1
70 WINDOW (-1.4,-1.2)-(1.8,1.2)
80 NMAX=4000
90 C=.5 : REM ***CONTRACTION FACTOR***
100 A=1/2 : B=SQR(3)/2 : REM ***TRANSFORMATION
    COEFFICIENTS***
110 X=1 : Y=0 : I=1 : REM ***START*
120 WHILE N<NMAX AND LEN(INKEY$)=0
130 R=1+INT(3*RND)
140 ON R GOSUB 200,220,240
150 X=X1 : Y=Y1
160 PSET (X,Y) : PSET (-A*X-B*Y,B*X-A*Y) :
    PSET (-A*X+B*Y,-B*X-A*Y)
170 PSET (X,-Y) : PSET (-A*X+B*Y,B*X+A*Y) :
    PSET (-A*X-B*Y,-B*X+A*Y)
180 N=N+1 : WEND : BEEP
190 A$=INPUT$(1) : END
200 X1=C*X-C+1 : Y1=C*Y : REM ***CONTRACTION WITH
    FIXED POINT 1,0 ***
210 RETURN
220 X1=C*X+(C-1)*A : Y1=C*Y-(C-1)*B : REM
    ***CONTRACTION WITH FIXED POINT -A,B ***
230 RETURN
240 X1=C*X+(C-1)*A : Y1=C*Y+(C-1)*B : REM
    CONTRACTION WITH FIXED POINT -A,-B ***
250 RETURN : END
```

```
10 REM ***THE MENGER SPONGE***
20 REM ***OBTAINED BY THE MONTE CARLO METHOD***
30 REM ***NAME:MENGERMC***
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS : RANDOMIZE 1
70 WINDOW (-1.6,-1.2)-(1.6,1.2)
80 X=1 : Y=1 : N=1 : NMAX=20000
90 WHILE N<NMAX AND LEN(INKEY$)=0
100 R=1+INT(8*RND)
110 ON R GOSUB 160,170,180,190,200,210,220,230
120 PSET (X,Y) : PSET (X,-Y) : PSET (-X,Y) :
    PSET (-X,-Y)
130 PSET (Y,X) : PSET (Y,-X) : PSET (-Y,X) :
    PSET (-Y,-X)
140 N=N+1 : WEND
150 BEEP : A$=INPUT$(1) : END
160 X=(X+2)/3 : Y=(Y+2)/3 : RETURN
170 X=(X+2)/3 : Y=(Y-2)/3 : RETURN
180 X=(X-2)/3 : Y=(Y+2)/3 : RETURN
190 X=(X-2)/3 : Y=(Y-2)/3 : RETURN
200 X=(X+2)/3 : Y=Y/3 : RETURN
210 X=(X-2)/3 : Y=Y/3 : RETURN
220 X=X/3 : Y=(Y+2)/3 : RETURN
230 X=X/3 : Y=(Y-2)/3 : RETURN
240 A$=INPUT$(1) : END
```

```
10 REM ***MENGER'S SPONGE IN COLOURS***
20 REM ***BACKTRACKING METHOD***
30 REM ***NAME:MENGERCL***
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS
70 P=4 : DIM X(8,P),Y(8,P),S(P),COL(P)
80 COL(1)=1 : COL(2)=4 : COL(3)=14 : COL(4)=10 :
   REM ***COLOUR CODE***
90 WINDOW (-1.6,-1.2)-(1.6,1.2)
100 LINE (-1,1)-(1,-1),COL(1),BF
110 LINE (-1/3,1/3)-(1/3,-1/3),COL(1),BF
120 X=0 : Y=0 : Q=0 : J=1 : S(0)=7
130 X1=(X+2)/3 : Y1=(Y+2)/3 : M=Q+J
140 X(2,M)=(X+2)/3 : Y(2,M)=(Y-2)/3
150 X(3,M)=(X-2)/3 : Y(3,M)=(Y+2)/3
160 X(4,M)=(X-2)/3 : Y(4,M)=(Y-2)/3
170 X(5,M)=(X+2)/3 : Y(5,M)=Y/3
180 X(6,M)=(X-2)/3 : Y(6,M)=Y/3
190 X(7,M)=X/3 : Y(7,M)=(Y+2)/3
200 X(8,M)=X/3 : Y(8,M)=(Y-2)/3
210 X=X1 : Y=Y1 : S(M)=7
220 H=3^(-M-1) : LINE (X1-H,Y1+H)-(X1+H,Y1-H),
    COL(M),BF
230 FOR L=2 TO 8
240 U=X(L,M) : V=Y(L,M)
250 U1=U-H : V1=V+H : U2=U+H : V2=V-H
260 LINE (U1,V1)-(U2,V2),COL(M),BF
270 NEXT L
280 IF M<P THEN J=J+1 : GOTO 130
290 K=1
300 IF S(M-K)=0 THEN K=K+1 : GOTO 300
310 IF M=K THEN GOTO 370
320 Q=M-K : X=X(2,Q) : Y=Y(2,Q)
330 FOR L=3 TO 8
340 X(L-1,Q)=X(L,Q) : Y(L-1,Q)=Y(L,Q)
350 NEXT L
360 S(Q)=S(Q)-1 : J=1 : GOTO 130
370 BEEP : A$=INPUT$(1) : END
```

```
10 REM ***PYTHAGORAS OR LEVY CURVE***
20 REM ***NAME:LEVYLINE***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS : PI=3.141593
60 WINDOW (-.7, -1.3)-(1.7,.5)
70 PRINT "SELECT ORDER P AS AN EVEN INTEGER <=14"
80 PRINT : INPUT P : CLS
90 H=2^(-(P/2)) : A=H*COS(P*PI/4 ): B=H*SIN(P*PI/4)
100 LINE (0,0)-(A,-B) : LINE -(A+B,A-B)
110 X=1 : Y=1
120 FOR N=2 TO 2^P-1
130 M=N : S=1
140 IF M MOD 2=1 THEN S=(S+1) MOD 4
150 M=M\2
160 IF M>1 THEN GOTO 140
170 IF S=0 THEN X=X+1
180 IF S=1 THEN Y=Y+1
190 IF S=2 THEN X=X-1
200 IF S=3 THEN Y=Y-1
210 LINE -(A*X+B*Y,A*Y-B*X)
220 NEXT N : BEEP
230 A$=INPUT$(1) : END
```

```
10 REM ***PYTHAGORAS OR LEVY CURVE***
20 REM ***NAME:LEVYLONG***
30 DEFLNG M,N,Q,R
40 SCREEN 12 : CLS : PI=3.141593
50 WINDOW (-.7, -1.3)-(1.7,.5)
60 PRINT "SELECT ORDER P AS AN EVEN INTEGER <=20"
70 PRINT : INPUT P : CLS
80 H=2^(-(P/2)) : A=H*COS(P*PI/4 ): B=H*SIN(P*PI/4)
90 LINE (0,0)-(A,-B) : LINE -(A+B,A-B)
100 X=1 : Y=1
110 FOR N=2 TO 2^P-1
120 M=N : S=0
130 FOR K=0 TO P
140 Q=INT(M/2) : R=M-2*Q : M=Q
150 S=S+R : NEXT K
160 S=S MOD 4
170 IF S=0 THEN X=X+1
180 IF S=1 THEN Y=Y+1
190 IF S=2 THEN X=X-1
200 IF S=3 THEN Y=Y-1
210 LINE -(A*X+B*Y,A*Y-B*X)
220 NEXT N : BEEP
230 A$=INPUT$(1) : END
```

```
10 REM ***DRAGON CURVE BETWEEN TWO POINTS***
20 REM ***NAME:DRAGON***
30 DEFLNG M,N,Q,R
40 SCREEN 12 : CLS : PI=3.141593
50 WINDOW (-.7,-1)-(1.7,.8)
60 INPUT"ORDER P = ",P
70 H=2^(-P/2) : S=0
80 X=H*COS(P*PI/4) : Y=H*SIN(P*PI/4)
90 CLS : LINE (0,0)-(X,Y)
100 FOR N=1 TO 2^P-1 : M=N
110 Q=INT(M/2) : R=M-2*Q
120 IF R=0 THEN M=M/2 : GOTO 110
130 Q=INT(M/4) : R=M-4*Q
140 IF R=1 THEN D=1 ELSE D=-1
150 S=(S+D) : F=(S-P/2)*PI/2
160 X=X+H*COS(F) : Y=Y+H*SIN(F)
170 LINE -(X,Y)
180 NEXT  N : A$=INPUT$(1) : END
```

```
10 REM ***GRAPH OF A MULTIFRACTAL BAR***
20 REM***NAME:MULTIFR***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 A=1/3 : B=1-A : P=12 : NMAX=2^P : H=1/NMAX
70 WINDOW (-.3,-.6)-(1.3,1.3)
80 LINE (0,0)-(1,0)
90 FOR N=0 TO NMAX-1 : S=0
100 M=N : FOR L=0 TO P-1
110 S=S+M MOD 2 : M=M\2 : NEXT L
120 X=H*N : Y=(A/B)^(P-S)
130 LINE (X,0)-(X,Y)
140 NEXT N : BEEP
150 A$=INPUT$(1) : END
```

Hoofdstuk 3

```
10 REM ***FRACTAL IN THE FORM OF A FERN***
20 REM ***NAME:FERN1***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS : RANDOMIZE 1
60 WINDOW (-20,-3)-(20,27)
70 COL=2 : REM ***SELECT COLOUR VALUE***
80 Q1=.8 : Q2=.9 : REM ***PROBABILITY DISTRIBUTION***
90 X=0 : Y=0 : N=0 : NMAX=10000
100 WHILE N<NMAX AND LEN(INKEY$)=0 : R=RND
110 IF R<Q1 THEN GOSUB 170
120 IF R>=Q1 AND R<Q2 THEN GOSUB 180
130 IF R>=Q2 THEN GOSUB 190
140 IF N>32 THEN PSET (X,Y),COL
150 N=N+1 : WEND
160 BEEP : A$=INPUT$(1) : END
170 X=11*X/12 : Y=11*Y/12+2 : RETURN
180 Z=X : X=Y/6 : Y=-Z/6+3 : RETURN
190 Z=X : X=-Y/6 : Y=Z/6+3 : RETURN
200 END
```

```
10 REM ***FRACTAL IN THE FORM OF A FERN***
20 REM ***DATA TAKEN FROM BARNSLEY'S BOOK
   (FRACTALS EVERYWHERE)***
30 REM ***NAME:FERN2***
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS : RANDOMIZE 1
70 WINDOW (-8,-1)-(8,11)
80 NMAX=80000! : COL=2
90 REM ***TABLE OF COEFFICIENTS***
100 D1=.16
110 A2=.85 : B2=.04 : C2=-.04 : D2=.85 : F2=1.6
120 A3=.2 : B3=-.26 : C3=.23 : D3=.22 : F3=1.6
130 A4=-.15 : B4=.28 : C4=.26 : D4=.24 : F4=.44
140 Q1=.01 : Q2=.86 : Q3=.93 : REM ***PROBABILITY
    DISTRIBUTION***
150 X=0 : Y=0 : N=0 : REM ***START***
160 FOR N=0 TO 20000 : R=RND
170 IF INKEY$<>"" THEN END
180 IF R<Q1 THEN GOSUB 250
190 IF R>=Q1 AND R<Q2 THEN GOSUB 260
200 IF R>=Q2 AND R<Q3 THEN GOSUB 270
210 IF R>=Q3 THEN GOSUB 280
220 IF N>32 THEN PSET (X,Y),COL
230 NEXT N : BEEP
240 A$=INPUT$(1) : END
250 X=0 : Y=D1*Y : RETURN
260 Z=X : X=A2*X+B2*Y : Y=C2*Z+D2*Y+F2 : RETURN
270 Z=X : X=A3*X+B3*Y : Y=C3*Z+D3*Y+F3 : RETURN
280 Z=X : X=A4*X+B4*Y : Y=C4*Z+D4*Y+F4 : RETURN
290 END
```

```
10 REM ***FRACTAL BASED ON THREE CONTRACTIONS***
20 REM ***NAME:SQUARE1***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS : RANDOMIZE 1
60 WINDOW (-1.6,-1.2)-(1.6,1.2)
70 Q1=1/3 : Q2=2/3 : REM ***PROBABILITY NUMBERS***
80 X=0 : Y=0 : REM ***START***
90 FOR I=1 TO 20000 : R=RND
100 IF R<Q1 THEN GOSUB 150
110 IF R>=Q1 AND R<Q2 THEN GOSUB 160
120 IF R>=Q2 THEN GOSUB 170
130 IF I>16 THEN PSET (X,Y) : PSET (-X,Y)
140 NEXT I : BEEP : A$=INPUT$(1) : END
150 X=-X/2-1/2 : Y=-Y/2-1/2 : RETURN
160 X=-X/2+1/2 : Y=-Y/2-1/2 : RETURN
170 X=-X/2 : Y=-Y/2+1/2 : RETURN
180 END
```

```
10 REM ***SYMMETRIC VERSION IN A SQUARE***
20 REM ***WITH THIS PROGRAM A GREAT VARIETY OF
   FRACTALS CAN BE FORMED***
30 REM ***ALL FRACTALS ARE CONTAINED WITHIN A
   SQUARE***
40 REM ***NAME:SQUARE2***
50 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
60 PRINT : INPUT"SCREEN = ",SCR
70 SCREEN SCR : CLS : RANDOMIZE 1 : PI=3.141593
80 WINDOW (-4,-3.3)-(2,1.2) : CLS
90 PRINT" 0 = IDENTITY , 1 = ROTATION 90 ,
   2 = ROTATION 180 , 3 = ROTATION 270"
100 PRINT" 4,5,6,7 AS 0,1,2,3 WITH REFLECTION"
110 PRINT" SELECT A GROUP OF THREE NUMBERS FROM
    0,1,2,3,4,5,6,7"
120 PRINT : PRINT" A GOOD EXAMPLE IS 5,3,5"
130 INPUT T1,T2,T3 : CLS
140 T=T1 : L=1 : GOSUB 260 : T=T2 : L=2 : GOSUB 260 :
    T=T3 : L=3 : GOSUB 260
150 E(1)=-1/2 : F(1)=-1/2 : E(2)=1/2 : F(2)=-1/2 :
    E(3)=-1/2 : F(3)=1/2
160 Q1=1/3 : Q2=2/3
170 X=0 : Y=0
180 FOR I=1 TO 20000 : R=RND
190 IF INKEY$<>"" THEN END
200 IF R<Q1 THEN L=1
210 IF R>=Q1 AND R<Q2 THEN L=2
220 IF R>=Q2 THEN L=3
230 Z=X : X=A(L)*X+B(L)*Y+E(L) : Y=C(L)*Z+D(L)*Y+F(L)
240 IF I>16 THEN PSET (X,Y) : PSET (-X-2,Y) :
    PSET (X,-Y-2) : PSET (-X-2,-Y-2)
250 NEXT I : BEEP : A$=INPUT$(1) : END
260 IF T=0 THEN A(L)=1/2 : B(L)=0 : C(L)=0 :
    D(L)=1/2
270 IF T=1 THEN A(L)=0 : B(L)=-1/2 : C(L)=1/2 :
    D(L)=0
280 IF T=2 THEN A(L)=-1/2 : B(L)=0 : C(L)=0 :
    D(L)=-1/2
290 IF T=3 THEN A(L)=0 : B(L)=1/2 : C(L)=-1/2 :
    D(L)=0
300 IF T=4 THEN A(L)=-1/2 : B(L)=0 : C(L)=0 :
    D(L)=1/2
310 IF T=5 THEN A(L)=0 : B(L)=1/2 : C(L)=1/2 :
    D(L)=0
320 IF T=6 THEN A(L)=1/2 : B(L)=0 : C(L)=0 :
    D(L)=-1/2
330 IF T=7 THEN A(L)=0 : B(L)=-1/2 : C(L)=-1/2 :
    D(L)=0
340 RETURN : END
```

```
10 REM ***ITERATIVE SYSTEM OF A SCALING AND AN
   INVERSION***
20 REM ***NAME:INVERMC***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS : RANDOMIZE 1
60 N=32000 : REM ***NUMBER OF POINTS***
70 WINDOW (-.6,-1.8)-(4.2,1.8)
80 PI=4*ATN(1)
90 C=.9 : ALFA=-(PI/4) : REM ***PARAMETERS***
100 A=C*COS(ALFA): B=C*SIN(ALFA)
110 X=2 : Y=0 : REM ***STARTING POINT***
120 FOR I=1 TO N
130 IF RND<.9 THEN GOSUB 160 ELSE GOSUB 170
140 PSET (X,Y)
150 NEXT I : BEEP : A$=INPUT$(1) : END
160 Z=X : X=A*X-B*Y+(2-2*A) : Y=B*Z+A*Y-2*B : RETURN
170 U=X*X+Y*Y : X=X/U : Y=Y/U : RETURN
180 END
```

Hoofdstuk 4

```
10 REM ***THE LORENZ ATTRACTOR***
20 REM ***NAME:LORENZ1***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS : PI=3.141593
60 WINDOW (-50,0)-(50,50)
70 A=10 : B=8/3 : C=28 : D=.003 : X=-12.5 :
   Y=-17.4 : Z=26
80 FOR K=0 TO 3 : CS=COS(K*PI/5) : SS=SIN(K*PI/5)
90 FOR M=1 TO 10000
100 U=X : V=X*CS-Y*SS : W=X*SS+Y*CS
110 PSET (V+W,Z),14
120 X1=X+D*A*(Y-X) : Y1=Y+D*(X*(C-Z)-Y) :
    Z1=Z+D*(X*Y-B*Z)
130 X2=X1+D*A*(Y1-X1) : Y2=Y1+D*(X1*(C-Z1)-Y1) :
    Z2=Z1+D*(X1*Y1-B*Z1)
140 X=(X+X2)/2 : Y=(Y+Y2)/2 : Z=(Z+Z2)/2
150 NEXT M : A$=INPUT$(1) : CLS
160 NEXT K : END
```

```
10 REM ***THE LORENZ ATTRACTOR***
20 REM ***SHOW OF THE SENSITIVITY UPON INITIAL
   CONDITIONS***
30 REM ***SHOWN ARE THE ORBITS OF Q (80 OR LESS)
   POINTS***
40 REM ***INITIALLY CLOSE TOGETHER ENDING UP IN
   CHAOS***
50 REM ***THE PROGRAM STOPS BY PRESSING ANY KEY***
60 REM ***NAME:LORENZ2***
70 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
80 PRINT : INPUT"SCREEN = ",SCR
90 SCREEN SCR : CLS
100 WINDOW (-50,0)-(50,50)
110 A=10 : B=8/3 : C=28 : D=.002 : Q=80
120 DIM X(Q),Y(Q),Z(Q)
130 FOR K=1 TO Q : X(K)=-12.5+K/100 : Y(K)=-17.4 :
    Z(K)=26 : NEXT K
140 WHILE INKEY$=""
150 FOR J=1 TO Q : X=X(J) : Y=Y(J) : Z=Z(J)
160 X1=X+D*A*(Y-X) : Y1=Y+D*(X*(C-Z)-Y) :
    Z1=Z+D*(X*Y-B*Z)
170 X2=X1+D*A*(Y1-X1) : Y2=Y1+D*(X1*(C-Z1)-Y1) :
    Z2=Z1+D*(X1*Y1-B*Z1)
180 X=(X+X2)/2 : Y=(Y+Y2)/2 : Z=(Z+Z2)/2
190 PSET (X+Y,Z),14 : PSET (X(J)+Y(J),Z(J)),0
200 X(J)=X : Y(J)=Y : Z(J)=Z
210 NEXT J
220 WEND : END
```

```
10 REM ***THE ROESSLER ATTRACTOR***
20 REM ***THE ATTRACTOR OF ROESSLER'S SYSTEM IS SHOWN
   IN PROJECTION***
30 REM ***THE PROGRAM IS STOPPED BY PRESSING ANY
   KEY***
40 REM ***NAME:ROESSLER***
50 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
60 PRINT : INPUT"SCREEN = ",SCR
70 SCREEN SCR : CLS : PI=3.141593
80 WINDOW (-24,-18)-(24,18)
90 PRINT"GIVE A VALUE OF C, GOOD VALUES ARE BETWEEN
   3.5 AND 4.5"
100 INPUT"C = ",C : CLS
110 PHI=45 : REM ***VIEWING ANGLE IN DEGREES***
120 P=COS(PHI*PI/180) : Q=SIN(PHI*PI/180)
130 A=.2 : B=.2 : D=.003 : X=4 : Y=-6 : Z=0
140 WHILE INKEY$=""
150 PSET (X,P*Y+Q*Z),14
160 X1=X-D*(Y+Z) : Y1=Y+D*(X+B*Y) :
    Z1=Z+D*(A+Z*(X-C))
170 X2=X1-D*(Y1+Z1) : Y2=Y1+D*(X1+B*Y1) :
    Z2=Z1+D*(A+Z1*(X-C))
180 X=(X+X2)/2 : Y=(Y+Y2)/2 : Z=(Z+Z2)/2
190 WEND : A$=INPUT$(1) : CLS
200 END
```

```
10 REM ***MODEL OF VERHULST***
20 REM ***NAME:VERHULST***
30 CLS : PRINT"GIVE SCREEN VALUE, 9 FOR EGA,
   12 FOR VGA OR OTHERWISE"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 PRINT"GIVE VALUE OF A"
70 INPUT"A = ",A
80 X=.5
90 FOR K=1 TO 200
100 IF K>1 AND K=20*INT(K/20)+1 THEN A$=INPUT$(1)
110 PRINT K,X
120 X=A*X*(1-X)
130 NEXT K
140 END
```

```
10 REM ***BIFURCATION DIAGRAM OF X'=A X(1-X)) ***
20 REM ***NAME:BIFUR1***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 INPUT"SCREEN = ",SCR
50 IF SCR=9 THEN YM=175
60 IF SCR=12 THEN YM=240
70 SCREEN SCR
80 F1=2*YM-80 : F2=F1-40
90 PRINT"GIVE VALUES OF A BETWEEN 3 AND 4"
100 INPUT"INITIAL VALUE A1 = ",A1
110 INPUT"FINAL VALUE A2 = ",A2
120 CLS : FOR N=0 TO 600
130 A=A1+N*(A2-A1)/600 : X=.5
140 FOR K=1 TO 400
150 X=A*X*(1-X)
160 IF K>200 THEN PSET(20+N,F1-F2*X)
170 NEXT K : NEXT N : END
```

```
10 REM ***BIFURCATION DIAGRAM OF X'=A X(1-Y),
   Y=AXY ***
20 REM ***TAKE 2.98<A<3.08 WITH .1<X<.5, OR***
30 REM ***TAKE 3.377<A<3.392 WITH 0<X<.7 ***
40 REM ***NAME:BIFUR2***
50 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
60 INPUT"SCREEN = ",SCR
70 IF SCR=9 THEN YM=175
80 IF SCR=12 THEN YM=240
90 SCREEN SCR : EPS=.01
100 WINDOW (0,0)-(639,.7)
110 PRINT"GIVE VALUES OF A"
120 INPUT"INITIAL VALUE A1 = ",A1
130 INPUT"FINAL VALUE A2 = ",A2
140 CLS : FOR N=0 TO 600
150 A=A1+N*(A2-A1)/600 : X=1/A : Y=1-2/A+EPS
160 FOR K=1 TO 800
170 Z=X : X=A*X*(1-X-Y) : Y=A*Z*Y
180 IF K>400 THEN PSET(20+N,X)
190 NEXT K : NEXT N : END
```

Hoofdstuk 5

```
10 REM ***ITERATIVE SYSTEM WITH TWO CONTRACTIONS***
20 REM ***SINGLE BOOKKEEPING***
30 REM ***NAME:TREESB***
40 SCREEN 9 : CLS : PI=4*ATN(1)
50 P=11 : DIM X(2^(P+1)-1),Y(2^(P+1)-1)
60 WINDOW (-2.4,-1.2)-(2.4,2.4)
70 R=SQR(1/2) : S=R : PHI=PI/4 : PSI=-PI/4
80 A=R*COS(PHI) : B=R*SIN(PHI) : C=S*COS(PSI) :
   D=S*SIN(PSI)
90 X(1)=0 : Y(1)=1 : PSET (X(1),Y(1))
100 FOR G=1 TO P : FOR N=2^(G-1) TO 2^G-1
110 X=X(N) : Y=Y(N)
120 GOSUB 160 : X(2*N)=X1 : Y(2*N)=Y1
130 GOSUB 180 : X(2*N+1)=X2 : Y(2*N+1)=Y2
140 PSET (X1,Y1) : PSET (X2,Y2)
150 NEXT N : NEXT G : A$=INPUT$(1) : END
160 X1=A*X-B*Y+A-1 : Y1=B*X+A*Y+B : REM ***
    CONTRACTION L**
170 RETURN
180 X2=C*X-D*Y+1-C : Y2=D*X+C*Y-D : REM ***
    CONTRACTION R***
190 RETURN
200 END
```

```
10 REM ***ITERATIVE SYSTEM WITH TWO CONTRACTIONS***
20 REM ***BACKTRACKING***
30 REM ***NAME:TREEBT***
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS
70 P=12 : DIM X(P),Y(P),Z(P)
80 WINDOW (-2.4,-1.2)-(2.4,2.4)
90 A=1/2 : B=1/2 : C=1/2 : D=-1/2
100 X=0 : Y=1 : REM : ***START***
110 PSET (X,Y) : M=0 : G=0
120 IF G<P-1 THEN GOSUB 170 ELSE GOSUB 200
130 IF M>=0 THEN GOTO 120 ELSE A$=INPUT$(1) : END
140 X1=A*X-B*Y+A-1 : Y1=B*X+A*Y+B : REM ***
    CONTRACTION L***
150 X2=C*X-D*Y+1-C : Y2=D*X+C*Y-D : REM ***
    CONTRACTION R***
160 PSET (X1,Y1) : PSET (X2,Y2) : RETURN
170 M=M+1 : G=G+1 : GOSUB 140
180 X=X1 : Y=Y1
190 X(M)=X2 : Y(M)=Y2 : Z(M)=G : RETURN
200 G=G+1 : GOSUB 140
210 X=X(M) : Y=Y(M) : M=M-1
220 GOSUB 140
230 X=X(M) : Y=Y(M) : G=Z(M)
240 M=M-1 : RETURN
250 END
```

```
10 REM ***ITERATIVE SYSTEM WITH TWO CONTRACTIONS***
20 REM ***CONDITIONAL BACKTRACKING***
30 REM ***NAME:TREEBTA***
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS
70 EPS=.01 : P=24 : DIM X(P),Y(P),Z(P),F(P)
80 WINDOW (-2.4,-1.2)-(2.4,2.4)
90 A1=1/2 : B1=1/2 : A2=1/3 : B2=-1/3
100 F1=SQR(A1*A1+B1*B1) : F2=SQR(A2*A2+B2*B2)
110 X=0 : Y=1 : F=1 : REM : ***START***
120 PSET (X,Y) : M=0 : G=0
130 IF G<P AND F>EPS THEN GOSUB 150 ELSE GOSUB 250
140 IF M>=0 THEN GOTO 130 ELSE A$=INPUT$(1) : END
150 X1=A1*X-B1*Y+A1-1 : Y1=B1*X+A1*Y+B1
160 X2=A2*X-B2*Y+1-A2 : Y2=B2*X+A2*Y-B2
170 PSET (X1,Y1) : PSET (X2,Y2)
180 REM ***DATA SECOND POINT TO STACK***
190 M=M+1 : G=G+1
200 X(M)=X2 : Y(M)=Y2 : Z(M)=G : F(M)=F*F2
210 REM ***GO ON WITH FIRST POINT***
220 X=X1 : Y=Y1 : F=F*F1
230 RETURN
240 REM ***GET DATA FROM STACK***
250 X=X(M) : Y=Y(M) : G=Z(M) : F=F(M) : M=M-1
260 RETURN
270 END
```

```
10 REM ***ITERATIVE SYSTEM WITH THREE CONTRACTIONS***
20 REM ***BACKTRACKING***
30 REM ***SPECIAL CASE OF THE SIERPINSKI SIEVE***
40 REM ***NAME:SIERBT***
50 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
60 PRINT : INPUT"SCREEN = ",SCR
70 SCREEN SCR : CLS
80 P=6 : DIM X(2*P),Y(2*P),Z(2*P)
90 WINDOW (-1.4,-1.2)-(1.8,1.2)
100 C=.5 : REM ***CONTRACTION FACTOR***
110 A=1/2 : B=SQR(3)/2 : REM ***TRANSFORMATION
    COEFFICIENTS***
120 X=0 : Y=0 : REM ***START***
130 PSET (X,Y) : M=0 : G=0 : R=.04
140 IF G<P-1 THEN GOSUB 210 ELSE GOSUB 260
150 IF M>=0 THEN GOTO 140 ELSE A$=INPUT$(1) : END
160 X1=C*X-C+1 : Y1=C*Y : REM ***CONTRACTION
    W.R.T. 1,0 ***
170 X2=C*X+(C-1)*A : Y2=C*Y-(C-1)*B : REM ***
    CONTRACTION W.R.T. -A,B ***
180 X3=C*X+(C-1)*A : Y3=C*Y+(C-1)*B : REM ***
    CONTRACTION W.R.T. -A,-B ***
190 PSET (X1,Y1) : PSET (X2,Y2) : PSET (X3,Y3)
200 RETURN
210 G=G+1 : GOSUB 160
220 X=X1 : Y=Y1
230 X(M+1)=X2 : Y(M+1)=Y2 : Z(M+1)=G
240 X(M+2)=X3 : Y(M+2)=Y3 : Z(M+2)=G
250 M=M+2 : RETURN
260 G=G+1 : GOSUB 160
270 X=X(M) : Y=Y(M) : M=M-1
280 GOSUB 160
290 X=X(M) : Y=Y(M) : M=M-1
300 GOSUB 160
310 X=X(M) : Y=Y(M) : G=Z(M)
320 M=M-1 : RETURN
330 END
```

Hoofdstuk 6

```
10 REM ***FRACTAL TENNIS***
20 REM ***NAME:TENNIS***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS : RANDOMIZE 1
60 WINDOW (-1.2,-.6)-(1.2,1.2)
70 X=RND : Y=RND : REM ***START*
80 FOR N=1 TO 6000
90 R=RND
100 IF R<1/3 THEN GOSUB 150 : GOTO 130
110 IF R>=1/3 AND R<2/3 THEN GOSUB 170 : GOTO 130
120 IF R>=2/3 THEN GOSUB 190
130 PSET (X,Y)
140 NEXT N : END
150 X=(X-1)/2 : Y=Y/2 : REM ***PLAYER A ***
160 RETURN
170 X=(X+1)/2 : Y=Y/2 : REM ***PLAYER B ***
180 RETURN
190 X=X/2 : Y=(Y+1)/2 : REM ***PLAYER C ***
200 RETURN : END
```

```
10 REM ***ITERATIVE SYSTEM WITH TWO CONTRACTIONS***
20 REM ***MONTE CARLO METHOD***
30 REM ***NAME:CONTRMC1***
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS
70 RANDOMIZE 1 : NMAX=10000
80 WINDOW (-3.2,-2.4)-(3.2,2.4)
90 A1=.64 : B1=.64 : A2=.1 : B2=-.1
100 F1=SQR(A1*A1+B1*B1) : F2=SQR(A2*A2+B2*B2)
110 X=0 : Y=1 : REM : ***START***
120 Q=F1/(F1+F2)
130 FOR N=1 TO NMAX
140 R=RND
150 IF R<Q THEN GOSUB 180 ELSE GOSUB 190
160 IF N>32 THEN  PSET (X,Y)
170 NEXT N : END
180 Z=X : X=A1*X-B1*Y+A1-1 : Y=B1*Z+A1*Y+B1 : RETURN
190 Z=X : X=A2*X-B2*Y+1-A2 : Y=B2*Z+A2*Y-B2 : RETURN
200 END
```

```
10 REM ***FRACTAL IN THE FORM OF A MAPLE LEAF***
20 REM ***DATA TAKEN FROM BARNSLEY'S BOOK
   (FRACTALS EVERYWHERE)***
30 REM ***NAME:MAPLE***
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS : RANDOMIZE 1
70 WINDOW (-.35,-.1)-(1.25,1.1)
80 NMAX=60000 : COL=2
90 REM ***TABLE OF COEFFICIENTS***
100 A1=.6 : B1=0 : C1=0 : D1=.6 : E1=.18 : F1=.36
110 A2=.6 : B2=0 : C2=0 : D2=.6 : E2=.18 : F2=.12
120 A3=.4 : B3=.3 : C3=-.3 : D3=.4 : E3=.27 : F3=.36
130 A4=.4 : B4= -.3 : C4=.3 : D4=.4 : E4=.27 : F4=.09
140 Q1=.3 : Q2=.6 : Q3=.8 : REM ***PROBABILITY
    DISTRIBUTION***
150 X=0 : Y=0 : N=0 : REM ***START***
160 WHILE N<NMAX AND LEN(INKEY$)=0 : R=RND
170 IF R<Q1 THEN GOSUB 240
180 IF R>=Q1 AND R<Q2 THEN GOSUB 250
190 IF R>=Q2 AND R<Q3 THEN GOSUB 260
200 IF R>=Q3 THEN GOSUB 270
210 IF N>32 THEN PSET (X,Y),COL
220 N=N+1 : WEND : BEEP
230 A$=INPUT$(1) : END
240 Z=X : X=A1*X+B1*Y+E1 : Y=C1*Z+D1*Y+F1 : RETURN
250 Z=X : X=A2*X+B2*Y+E2 : Y=C2*Z+D2*Y+F2 : RETURN
260 Z=X : X=A3*X+B3*Y+E3 : Y=C3*Z+D3*Y+F3 : RETURN
270 Z=X : X=A4*X+B4*Y+E4 : Y=C4*Z+D4*Y+F4 : RETURN
280 END
```

```
10 REM ***COLLAGE OF LETTER H***
20 REM ***NAME:COLLAGEH***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS : RANDOMIZE 100
60 WINDOW (-6,-4)-(6,4)
70 X=0 : Y=0
80 FOR N=1 TO 16000
90 R=INT(7*RND)+1
100 ON R GOSUB 130,140,150,160,170,180,190
110 GOSUB 200
120 NEXT N : END
130 X=X/3+2 : Y=Y/3+2 : RETURN
140 X=X/3-2 : Y=Y/3+2 : RETURN
150 X=X/3-2 : Y=Y/3-2 : RETURN
160 X=X/3+2 : Y=Y/3-2 : RETURN
170 X=X/3+2 : Y=Y/3 : RETURN
180 X=X/3-2 : Y=Y/3 : RETURN
190 X=X/3 : Y=Y/3 : RETURN
200 PSET (X,Y) : PSET (X,-Y) : PSET (-X,Y) :
    PSET (-X,-Y)
210 RETURN : END
```

Hoofdstuk 7

```
10 REM ***RANDOM WALK IN A RECTANGLE***
20 REM ***NAME:RWALK***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 RANDOMIZE 123
70 IF SCR=9 THEN XM=320 : YM=175
80 IF SCR=12 THEN XM=320 : YM=240
90 A=200 : B=150 : REM ***THERE ARE 2A+1 COLUMNS AND
   2B+1 ROWS***
100 INPUT"AANTAL STAPPEN = ",N
110 INPUT"STAPLENGTE     = ",H
120 CLS : LINE (XM-A,YM-B)-(XM-A,YM+B) :
    LINE -(XM+A,YM+B)
130 LINE -(XM+A,YM-B) : LINE -(XM-A,YM-B)
140 X=XM : Y=YM : PSET (XM,YM)
150 FOR K=1 TO N
160 R=INT(4*RND)
170 IF R=0 THEN X1=X+H : Y1=Y
180 IF R=1 THEN X1=X-H : Y1=Y
190 IF R=2 THEN X1=X : Y1=Y-H
200 IF R=3 THEN X1=X : Y1=Y+H
210 IF ABS(X1-XM)>A THEN END
220 IF ABS(Y1-YM)>B THEN END
230 LINE -(X1,Y1)
240 X=X1 : Y=Y1
250 NEXT K : END
```

```
10 REM ***CIRCULAR AGGREGATION***
20 REM ***NAME:CLUSTERC***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 XM=320 : PI=4*ATN(1)
70 IF SCR=12 THEN YM=240 ELSE YM=175
80 PSET (XM,YM),14 : K=0 : MAX=0
90 RANDOMIZE 11
100 WHILE K<8000 AND LEN(INKEY$)=0
110 K=K+1 : LOCATE 1,1 : PRINT K
120 R=MAX*(1+RND/5)
130 C=2*PI*RND : P=XM+INT(R*COS(C)) :
    Q=YM+INT(R*SIN(C))
140 S=1+INT(4*RND)
150 ON S GOSUB 240,250,260,270
160 C1=POINT (P-1,Q) : C2=POINT (P+1,Q)
170 C3=POINT (P,Q-1) : C4=POINT (P,Q+1)
180 D1=POINT (P-1,Q-1) : D2=POINT (P+1,Q+1)
190 D3=POINT (P-1,Q+1) : D4=POINT (P+1,Q-1)
200 IF C1=14 OR C2=14 OR C3=14 OR C4=14
    THEN GOTO 280
210 IF D1=14 OR D2=14 OR D3=14 OR D4=14
    THEN GOTO 280
220 IF ABS(P-XM)>MAX+4 OR ABS(Q-YM)>MAX+4
    THEN GOTO 130
230 GOTO 140
240 P=P-1 : RETURN
250 P=P+1 : RETURN
260 Q=Q-1 : RETURN
270 Q=Q+1 : RETURN
280 PSET (P,Q),14
290 IF MAX<ABS(P-XM) THEN MAX=ABS(P-XM)
300 IF MAX<ABS(Q-YM) THEN MAX=ABS(Q-YM)
310 WEND : BEEP : A$=INPUT$(1) : END
```

```
10 REM ***AGGREGATION ON A LINE IN A RECTANGLE***
20 REM ***NAME:CLUSTERL***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 IF SCR=12 THEN YM=400 ELSE YM=300
70 XM=320 : K=0 : MAX=0
80 RANDOMIZE 11
90 WHILE K<20000 AND LEN(INKEY$)=0
100 K=K+1 : LOCATE 1,1 : PRINT K
110 P=XM+INT((RND-.5)*300) : Q=YM-MAX-8
120 S=1+INT(4*RND)
130 ON S GOSUB 220,240,260,280
140 C1=POINT (P-1,Q) : C2=POINT (P+1,Q)
150 C3=POINT (P,Q-1) : C4=POINT (P,Q+1)
160 D1=POINT (P+1,Q+1) : D2=POINT (P-1,Q-1)
170 D3=POINT (P-1,Q+1) : D4=POINT (P+1,Q-1)
180 IF C1=14 OR C2=14 OR C3=14 OR C4=14
    THEN GOTO 300
190 IF D1=14 OR D2=14 OR D3=14 OR D4=14
    THEN GOTO 300
200 IF Q=YM THEN PSET (P,Q),14 : GOTO 320
210 GOTO 120
220 P=P-1 : IF P<170 THEN P=P+2
230 RETURN
240 P=P+1 : IF P>470 THEN P=P-2
250 RETURN
260 Q=Q-1 : IF Q<YM-MAX-12 THEN Q=Q+2
270 RETURN
280 Q=Q+1
290 RETURN
300 PSET (P,Q),14
310 IF MAX<YM-Q THEN MAX=YM-Q
320 WEND : BEEP
330 A$=INPUT$(1) : END
```

Hoofdstuk 8

```
10 REM ***KOCH LINE AFTER LINDENMAYER***
20 REM ***NAME:KOCHLIND***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 PRINT : PRINT"GIVE ORDER UP TO 5"
60 PRINT : INPUT"ORDER = ",GMAX
70 SCREEN SCR : CLS
80 WINDOW (-.3,-.5)-(1.3,.7)
90 DIM X(2048),Y(2048)
100 NR=1 : X(1)=0 : PI=4*ATN(1)
110 FOR G=1 TO GMAX : P=0
120 FOR I=1 TO NR : S=X(I)
130 IF S=0 THEN GOSUB 290
140 IF S=1 THEN P=P+1 : Y(P)=1
150 IF S=2 THEN P=P+1 : Y(P)=2
160 IF S=3 THEN P=P+1 : Y(P)=3
170 NEXT I  : NR=P
180 FOR I=1 TO NR : X(I)=Y(I) : NEXT I
190 NEXT G
200 REM ***DRAWING PROCEDURE***
210 H=3^(-GMAX) : F=0 : PSET (0,0),14
220 FOR I=1 TO NR : S=X(I)
230 IF S=1 THEN F=F+PI/3 : GOTO 280
240 IF S=2 THEN F=F-2*PI/3 : GOTO 280
250 IF S=3 THEN F=F+PI/3 : GOTO 280
260 U=U+H*COS(F) : V=V+H*SIN(F)
270 LINE -(U,V),14
280 NEXT I : A$=INPUT$(1) : END
290 Y(P+1)=0 : Y(P+2)=1 : Y(P+3)=0
300 Y(P+4)=2 : Y(P+5)=0 : Y(P+6)=3
310 Y(P+7)=0 : P=P+7 : RETURN
320 END
```

```
10 REM ***KOCH LINE AFTER LINDENMAYER***
20 REM ***NAME:SIERLIND***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 PRINT : PRINT"GIVE ORDER UP TO 5"
60 PRINT : INPUT"ORDER = ",GMAX
70 SCREEN SCR : CLS
80 WINDOW (-2,-1)-(2,2)
90 DIM X(2000),Y(2000)
100 NR=1 : X(1)=1 : PI=4*ATN(1)
110 FOR G=1 TO GMAX : P=0
120 FOR I=1 TO NR : S=X(I)
130 IF S=0 THEN GOSUB 290
140 IF S=1 THEN GOSUB 310
150 IF S=2 THEN P=P+1 : Y(P)=2
160 IF S=3 THEN P=P+1 : Y(P)=3
170 NEXT I  : NR=P
180 FOR I=1 TO NR : X(I)=Y(I) : NEXT I
190 NEXT G
200 REM ***DRAWING PROCEDURE***
210 H=2^(-GMAX) : F=0 : PSET (0,0),14
220 FOR I=1 TO NR
230 IF X(I)=1 THEN GOTO 280
240 IF X(I)=2 THEN F=F+2*PI/3 : GOTO 280
250 IF X(I)=3 THEN F=F-2*PI/3 : GOTO 280
260 U=U+H*COS(F) : V=V+H*SIN(F)
270 LINE -(U,V),14
280 NEXT I : A$=INPUT$(1) : END
290 Y(P+1)=0 : Y(P+2)=0
300 P=P+2 : RETURN
310 Y(P+1)=2 : Y(P+2)=0 : Y(P+3)=1 : Y(P+4)=0
320 Y(P+5)=3 : Y(P+6)=0 : Y(P+7)=1 : Y(P+8)=0
330 Y(P+9)=3 : Y(P+10)=0 : Y(P+11)=1 :
    Y(P+12)=0 : Y(P+13)=2
340 P=P+13 : RETURN
350 END
```

```
10 REM ***PEANO LINE AFTER LINDENMAYER***
20 REM ***NAME:PEANOL***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 PRINT : PRINT"GIVE ORDER UP TO 5"
60 PRINT : INPUT"ORDER = ",GMAX
70 SCREEN SCR : CLS
80 WINDOW (-.7,-.5)-(1.7,1.3)
90 DIM X(3411),Y(3411)
100 NR=1 : X(1)=1 : PI=4*ATN(1)
110 FOR G=1 TO GMAX : P=0
120 FOR I=1 TO NR : S=X(I)
130 IF S=0 THEN P=P+1 : Y(P)=0
140 IF S=1 THEN GOSUB 300
150 IF S=2 THEN GOSUB 350
160 IF S=3 THEN P=P+1 : Y(P)=3
170 IF S=4 THEN P=P+1 : Y(P)=4
180 NEXT I  : NR=P
190 FOR I=1 TO NR : X(I)=Y(I) : NEXT I
200 NEXT G
210 REM ***DRAWING PROCEDURE***
220 H=2^(-GMAX) : F=0 : PSET (0,0),14
230 FOR I=1 TO NR : S=X(I)
240 IF S=3 THEN F=F-PI/2 : GOTO 290
250 IF S=4 THEN F=F+PI/2 : GOTO 290
260 IF S=1 OR S=2 THEN GOTO 290
270 U=U+H*COS(F) : V=V+H*SIN(F)
280 LINE -(U,V),14
290 NEXT I : A$=INPUT$(1) : END
300 Y(P+1)=4 : Y(P+2)=2 : Y(P+3)=0
310 Y(P+4)=3 : Y(P+5)=1 : Y(P+6)=0
320 Y(P+7)=1 : Y(P+8)=3 : Y(P+9)=0
330 Y(P+10)=2 : Y(P+11)=4
340 P=P+11 : RETURN
350 Y(P+1)=3 : Y(P+2)=1 : Y(P+3)=0
360 Y(P+4)=4 : Y(P+5)=2 : Y(P+6)=0
370 Y(P+7)=2 : Y(P+8)=4 : Y(P+9)=0
380 Y(P+10)=1 : Y(P+11)=3
390 P=P+11 : RETURN
400 END
```

```
10 REM ***A SYNTHETIC TREE AFTER LINDENMAYER***
20 REM ***NAME:BUSHLIND***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 PRINT : PRINT"GIVE ORDER UP TO 5"
60 PRINT : INPUT"ORDER = ",GMAX
70 SCREEN SCR : CLS
80 WINDOW (-4,-1)-(4,5)
90 DIM X%(6400),Y%(6400)
100 NR=1 : X(1)=0 : PI=4*ATN(1)
110 FOR G=1 TO GMAX : P=0
120 FOR I=1 TO NR : S=X%(I)
130 IF S=0 THEN GOSUB 320 ELSE P=P+1 : Y%(P)=S
140 NEXT I : NR=P
150 FOR I=1 TO NR : X%(I)=Y%(I) : NEXT I
160 NEXT G
170 REM ***DRAWING PROCEDURE***
180 XA=0 : YA=0 : FA=PI/2 : H=2^(-G) : M=0 : COL=14
190 FOR I=1 TO NR : S=X%(I)
200 IF S=0 THEN GOSUB 250
210 IF S=1 THEN GOSUB 280
220 IF S=2 THEN GOSUB 290
230 IF S=3 THEN GOSUB 300
240 NEXT I : A$=INPUT$(1) : END
250 XB=XA+H*COS(FA) : YB=YA+H*SIN(FA)
260 LINE (XA,YA)-(XB,YB),COL
270 XA=XB : YA=YB : RETURN
280 M=M+1 : U(M)=XA : V(M)=YA : W(M)=FA :
    FA=FA+PI/5 : RETURN
290 M=M+1 : U(M)=XA : V(M)=YA : W(M)=FA :
    FA=FA-PI/5 : RETURN
300 XA=U(M) : YA=V(M) : FA=W(M) : M=M-1
310 CIRCLE (XB,YB),.01,4 : RETURN
320 Y%(P+1)=0 : Y%(P+2)=1 : Y%(P+3)=0
330 Y%(P+4)=3 : Y%(P+5)=0 : Y%(P+6)=2
340 Y%(P+7)=0 : Y%(P+8)=3 : Y%(P+9)=0
350 P=P+9 : RETURN
360 END
```

Hoofdstuk 9

```
10 REM ***A GROWTH MODEL BY PICKOVER***
20 REM ***SYMMETRIC VERSION***
30 REM***NAME:PICK1***
40 DEFINT I,J,K,N,T,X,Y
50 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
60 INPUT"SCREEN = ",SCR
70 IF SCR=9 THEN XM=320 : YM=175
80 IF SCR=12 THEN XM=320 : YM=240
90 SCREEN SCR : CLS
100 INPUT"NUMBER OF ROWS = ",N
110 REM ***TOTAL NUMBER OF ROWS IS 2N-3 ***
120 CLS : DIM X(N,N),Y(N,N)
130 X(0,0)=1
140 FOR K=1 TO N-1
150 FOR I=0 TO K : FOR J=0 TO K-I
160 IF X(I,J)=0 THEN GOSUB 220 ELSE GOSUB 270
170 NEXT J : NEXT I
180 FOR I=0 TO K : FOR J=0 TO K-I
190 X(I,J)=Y(I,J)
200 NEXT J : NEXT I : NEXT K
210 A$=INPUT$(1) : END
220 IF I>=1 AND J>=1 THEN T=X(I+1,J)+X(I-1,J)+
    X(I,J+1)+X(I,J-1)
230 IF I=0 AND J>=1 THEN T=2*X(1,J)+X(0,J+1)+
    X(0,J-1)
240 IF I>=1 AND J=0 THEN T=2*X(I,1)+X(I+1,0)+X(I-1,0)
250 IF T=1 THEN Y(I,J)=1
260 RETURN
270 PSET (XM+2*I,YM-2*J),14 : PSET (XM-2*I,YM-2*J),14
280 PSET (XM+2*I,YM+2*J),14 : PSET (XM-2*I,YM+2*J),14
```

```
10 REM ***THE EVEN-ODD GROWTH MODEL OF PICKOVER***
20 REM ***CF ALGORITHM 1.3, 1990***
30 REM ***SYMMETRIC VERSION***
40 REM***NAME:PICK2***
50 DEFINT I,J,K,N,S,X,Y
60 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
70 INPUT"SCREEN = ",SCR
80 IF SCR=9 THEN XM=320 : YM=175
90 IF SCR=12 THEN XM=320 : YM=240
100 SCREEN SCR : CLS
110 INPUT"NUMBER OF ROWS = ",N
120 CLS : DIM X(N,N),Y(N,N)
130 X(0,0)=1
140 FOR K=1 TO N-1
150 FOR I=0 TO K : FOR J=0 TO K
160 IF X(I,J)=0 THEN GOSUB 220 ELSE GOSUB 340
170 NEXT J : NEXT I
180 FOR I=0 TO K : FOR J=0 TO K
190 X(I,J)=Y(I,J)
200 NEXT J : NEXT I : NEXT K
210 A$=INPUT$(1) : END
220 IF K MOD 2 = 1 THEN GOSUB 240 ELSE GOSUB 290
230 RETURN
240 IF I>=1 AND J>=1 THEN S=X(I+1,J)+X(I-1,J)+
    X(I,J+1)+X(I,J-1)
250 IF I=0 AND J>=1 THEN S=2*X(1,J)+X(0,J+1)+X(0,J-1)
260 IF I>=1 AND J=0 THEN S=2*X(I,1)+X(I+1,0)+X(I-1,0)
270 IF S=1 THEN Y(I,J)=1 ELSE Y(I,J)=0
280 RETURN
290 IF I>=1 AND J>=1 THEN S=X(I+1,J)+X(I-1,J)+
    X(I,J+1)+X(I,J-1)+X(I+1,J+1)+X(I-1,J+1)+
    X(I-1,J-1)+X(I+1,J-1)
300 IF I=0 AND J>=1 THEN S=2*X(1,J)+2*X(1,J+1)+
    2*X(1,J-1)+X(0,J+1)+X(0,J-1)
310 IF I>=1 AND J=0 THEN S=2*X(I,1)+2*X(I+1,1)+
    2*X(I-1,1)+X(I+1,0)+X(I-1,0)
320 IF S=1 THEN Y(I,J)=1 ELSE Y(I,J)=0
330 RETURN
340 PSET (XM+2*I,YM-2*J),14 : PSET (XM-2*I,YM-2*J),14
350 PSET (XM+2*I,YM+2*J),14 : PSET (XM-2*I,YM+2*J),14
360 RETURN : END
```

```
10 REM ***THE EVEN-ODD GROWTH MODEL OF PICKOVER***
20 REM ***CF ALGORITHM 1.3, 1990***
30 REM ***COLOUR VERSION***
40 REM***NAME:PICK3***
50 DEFINT I,J,K,N,S,X,Y
60 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
70 INPUT"SCREEN = ",SCR
80 IF SCR=9 THEN XM=320 : YM=175
90 IF SCR=12 THEN XM=320 : YM=240
100 SCREEN SCR : CLS
110 PRINT"DETERMINE NUMBER OF ROWS, TAKE N=128
    OR LOWER"
120 INPUT"NUMBER OF ROWS = ",N
130 CLS : DIM X(N,N),Y(N,N),KL(7)
140 KL(1)=4 : KL(2)=12 : KL(3)=2 : KL(4)=10
150 KL(5)=14 : KL(6)=5 : KL(7)=13
160 X(0,0)=1 : REM ***START***
170 FOR K=1 TO N-1
180 FOR I=0 TO K : FOR J=0 TO K
190 IF X(I,J)=0 THEN GOSUB 260 ELSE Y(I,J)=X(I,J)
200 NEXT J : NEXT I
210 FOR I=0 TO K : FOR J=0 TO K
220 IF K MOD 2 = 1 THEN GOSUB 380 : REM ***PRINT IN
    COLOUR***
230 IF Y(I,J)=1 THEN X(I,J)=1 ELSE X(I,J)=0
240 NEXT J : NEXT I : NEXT K
250 A$=INPUT$(1) : END
260 IF K MOD 2 = 1 THEN GOSUB 280 ELSE GOSUB 330
270 RETURN
280 IF I>=1 AND J>=1 THEN S=X(I+1,J)+X(I-1,J)+
    X(I,J+1)+X(I,J-1)
290 IF I=0 AND J>=1 THEN S=2*X(1,J)+X(0,J+1)+X(0,J-1)
300 IF I>=1 AND J=0 THEN S=2*X(I,1)+X(I+1,0)+X(I-1,0)
310 Y(I,J)=S
320 RETURN
330 IF I>=1 AND J>=1 THEN S=X(I+1,J)+X(I-1,J)+
    X(I,J+1)+X(I,J-1)+X(I+1,J+1)+X(I-1,J+1)+
    X(I-1,J-1)+X(I+1,J-1)
340 IF I=0 AND J>=1 THEN S=2*X(1,J)+2*X(1,J+1)+
    2*X(1,J-1)+X(0,J+1)+X(0,J-1)
350 IF I>=1 AND J=0 THEN S=2*X(I,1)+2*X(I+1,1)+
    2*X(I-1,1)+X(I+1,0)+X(I-1,0)
360 Y(I,J)=S
370 RETURN
380 C=KL(Y(I,J))
390 PSET (XM+2*I,YM-2*J),C : PSET (XM-2*I,YM-2*J),C
400 PSET (XM+2*I,YM+2*J),C : PSET (XM-2*I,YM+2*J),C
410 RETURN : END
```

```
10 REM ***CELLULAR AUTOMATON WITH SPECIAL RULE***
20 REM ***A NEW CELL IS DETERMINED BY 3 OLDER
   CELLS***
30 REM ***CELLS ARE BOUNDED BY ZEROS***
40 REM ***NAME:CELL1***
50 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
60 INPUT"SCREEN = ",SCR
70 IF SCR=9 THEN XM=320 : YM=175
80 IF SCR=12 THEN XM=320 : YM=240
90 SCREEN SCR
100 PRINT"GIVE HALF NUMBER OF CELLS AND HALF NUMBER
    OF ROWS"
110 PRINT : INPUT"HALF NUMBER OF CELLS = ",N
120 PRINT : INPUT"HALF NUMBER OF ROWS = ",P
130 DIM X(2*N),Y(2*N)
140 X(0)=0 : X(2*N)=0 : CLS
150 X(N)=1 : PSET (XM,YM-P)
160 FOR K=1 TO 2*P+1
170 FOR L=1 TO 2*N-1
180 Y(L)=(X(L-1)+X(L+1)) MOD 2 : REM ***PRODUCTION
    RULE***
190 NEXT L
200 FOR I=1 TO 2*N-1
210 X(I)=Y(I)
220 IF X(I)=1 THEN PSET (XM-N+I,YM-P+K)
230 NEXT I : NEXT K
240 A$=INPUT$(1) : END
```

```
10 REM ***CELLULAR AUTOMATON WITH SPECIAL RULE***
20 REM ***A NEW CELL IS DETERMINED BY 3 OLDER
   CELLS***
30 REM ***CELLS ARE BOUNDED BY ZEROS***
40 REM ***NAME:CELL2***
50 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
60 INPUT"SCREEN = ",SCR
70 IF SCR=9 THEN XM=320 : YM=175
80 IF SCR=12 THEN XM=320 : YM=240
90 IF (SCR<>9 AND SCR<>12) THEN BEEP : GOTO 50
100 SCREEN SCR
110 PRINT"GIVE HALF NUMBER OF CELLS AND HALF NUMBER
    OF ROWS"
120 PRINT : INPUT"HALF NUMBER OF CELLS = ",N
130 PRINT : INPUT"HALF NUMBER OF ROWS = ",P
140 DIM X(2*N),Y(2*N)
150 X(0)=0 : X(2*N)=0 : CLS
160 X(N)=1 : PSET (XM,YM-P),14
170 FOR K=1 TO 2*P+1
180 FOR L=1 TO 2*N-1
190 Y(L)=(X(L-1)+X(L)+X(L+1)) MOD 2 : REM ***
    PRODUCTION RULE***
200 NEXT L
210 FOR I=1 TO 2*N-1
220 X(I)=Y(I)
230 IF X(I)=1 THEN PSET (XM-N+I,YM-P+K),14
240 NEXT I : NEXT K
250 A$=INPUT$(1) : END
```

```
10 REM ***CELLULAR AUTOMATON WITH SPECIAL RULE***
20 REM ***A NEW CELL IS DETERMINED BY 3 OLDER
   CELLS***
30 REM ***CELLS ARE BOUNDED BY ZEROS***
40 REM ***START BY A RANDOM ROW***
50 REM ***NAME:CELL1A***
60 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
70 INPUT"SCREEN = ",SCR
80 IF SCR=9 THEN XM=320 : YM=175
90 IF SCR=12 THEN XM=320 : YM=240
100 SCREEN SCR : RANDOMIZE 1
110 PRINT"GIVE HALF NUMBER OF CELLS AND HALF NUMBER
    OF ROWS"
120 PRINT : INPUT"HALF NUMBER OF CELLS = ",N
130 PRINT : INPUT"HALF NUMBER OF ROWS = ",P
140 DIM X(2*N),Y(2*N)
150 X(0)=0 : X(1)=0 : X(2*N-1)=0 : X(2*N)=0 : CLS
160 FOR I=1 TO N-1
170 X(2*I)=INT(2*RND) : X(2*I-1)=0
180 NEXT I
190 FOR K=1 TO 2*P+1
200 FOR L=1 TO 2*N-1
210 Y(L)=(X(L-1)+X(L+1)) MOD 2 : REM ***PRODUCTION
    RULE***
220 NEXT L
230 FOR I=1 TO 2*N-1
240 X(I)=Y(I)
250 IF X(I)=1 THEN PSET (XM-N+I,YM-P+K)
260 NEXT I : NEXT K
270 A$=INPUT$(1) : END
```

```
10 REM ***CELLULAR AUTOMATON WITH GENERAL RULE***
20 REM ***A NEW CELL IS DETERMINED BY 3 OLDER
   CELLS***
30 REM ***CELLS ARE BOUNDED BY ZEROS***
40 REM ***NAME:CELL3***
50 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
60 INPUT"SCREEN = ",SCR
70 IF SCR=9 THEN XM=320 : YM=175
80 IF SCR=12 THEN XM=320 : YM=240
90 SCREEN SCR : CLS
100 PRINT"GIVE HALF NUMBER OF CELLS AND HALF NUMBER
    OF ROWS"
110 PRINT : INPUT"HALF NUMBER OF CELLS = ",N
120 PRINT : INPUT"HALF NUMBER OF ROWS = ",P
130 PRINT"GIVE PRODUCTION RULE AS INTEGER
    BETWEEN 0 AND 255***
140 PRINT"GOOD CASES ARE 18,22,54,90,126,146,150,210"
150 PRINT : INPUT"CODE = ",C
160 DIM X(2*N),Y(2*N),B(7) : CLS
170 M=C : FOR J=0 TO 7
180 Q=INT(M/2) : B(J)=M-2*Q : M=Q
190 NEXT J
200 X(0)=0 : X(N)=1 : X(2*N)=0 : PSET (XM,YM-P)
210 FOR K=1 TO 2*P+1
220 FOR L=1 TO 2*N-1
230 T=X(L+1)+2*X(L)+4*X(L-1)
240 Y(L)=B(T) : NEXT L
250 FOR I=1 TO 2*N-1
260 X(I)=Y(I)
270 IF X(I)=1 THEN PSET (XM-N+I,YM-P+K),14
280 NEXT I : NEXT K
290 A$=INPUT$(1) : END
```

```
10 REM ***CELLULAR AUTOMATON WITH GENERAL TOTALISTIC
   RULE***
20 REM ***A NEW CELL IS DETERMINED BY 5 OLDER
   CELLS***
30 REM ***CELLS ARE BOUNDED BY ZEROS***
40 REM ***NAME:CELLC***
50 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
60 PRINT : INPUT"SCREEN = ",SCR
70 IF SCR=9 THEN XM=320 : YM=175
80 IF SCR=12 THEN XM=320 : YM=240
90 SCREEN SCR : CLS : RANDOMIZE 1
100 PRINT"GIVE HALF NUMBER OF CELLS AND HALF NUMBER
    OF ROWS"
110 PRINT : INPUT"HALF NUMBER OF CELLS = ",N
120 PRINT : INPUT"HALF NUMBER OF ROWS = ",P : PRINT
130 PRINT"GIVE PRODUCTION RULE AS EVEN INTEGER
    BETWEEN 0 AND 64***
140 PRINT : INPUT"CODE = ",C
150 DIM X(2*N+2),Y(2*N+2),B(5) : CLS
160 M=C : FOR J=0 TO 5
170 Q=INT(M/2) : B(J)=M-2*Q : M=Q
180 NEXT J
190 X(0)=0 : X(1)=0 : X(2*N+1)=0 : X(2*N+2)=0 : CLS
200 FOR I=2 TO 2*N : X(I)=INT(2*RND)
210 IF X(I)=1 THEN PSET (XM-N+I-1,YM-P),14
220 IF INKEY$<>"" THEN END
230 NEXT I
240 FOR K=1 TO 2*P+1
250 FOR L=2 TO 2*N
260 S=X(L-2)+X(L-1)+X(L)+X(L+1)+X(L+2)
270 Y(L)=B(S) : NEXT L
280 FOR I=2 TO 2*N
290 X(I)=Y(I)
300 IF X(I)=1 THEN PSET (XM-N+I-1,YM-P+K),14
310 NEXT I : NEXT K
320 A$=INPUT$(1) : END
```

Hoofdstuk 10

```
10 REM ***3D BARE PYTHAGORAS TREE, BACKTRACKING***
20 REM ***FIXED ORDER***
30 REM ***NAME:PYT3DBT***
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 P=11 : DIM XA(P),YA(P),ZA(P),XB(P),YB(P),ZB(P),
   S(P),KL(P)
70 SCREEN SCR : CLS : PI=3.141593
80 WINDOW (-4,-2)-(4,4)
90 DATA 6,8,6,8,6,2,4,2,10,12,14,14
100 FOR I=1 TO 11 : READ KL(I) : NEXT I
110 H1=20 : H2=-60 : REM ***PRESCRIBED ANGLES***
120 H1=H1*PI/180 : H2=H2*PI/180
130 R1=.85 : R2=.7 : REM ***PRESCRIBED REDUCTIONS***
140 A1=R1*COS(H1) : B1=R1*SIN(H1)
150 A2=R2*COS(H2) : B2=R2*SIN(H2)
160 PHI=10 : PHI=PHI*PI/180 : REM ***VIEWING
    ANGLE***
170 CS=COS(PHI) : SS=SIN(PHI)
180 G=0 : M=0
190 XA=.01 : YA=0 : ZA=-1 : XB=0 : YB=0 : ZB=0
200 UA=YA : VA=CS*ZA+SS*XA : UB=XB : VB=CS*ZB+SS*YB
210 LINE (UA,VA)-(UB,VB),6
220 IF G<P-1 THEN GOSUB 350 ELSE GOSUB 410
230 IF M>=0 THEN GOTO 220 ELSE A$=INPUT$(1) : END
240 U=XB-XA : V=YB-YA : W=ZB-ZA
250 T=SQR(1+W*W/(U*U+V*V))
260 XX=XB : YY=YB : ZZ=ZB
270 X1=XB+A1*U-T*B1*V : Y1=YB+T*B1*U+A1*V :
    Z1=ZB+A1*W
280 X2=XB+A2*U-T*B2*V : Y2=YB+T*B2*U+A2*V :
    Z2=ZB+A2*W
290 RETURN
300 UB=YB : VB=CS*ZB+SS*XB
310 U1=Y1 : V1=CS*Z1+SS*X1
320 U2=Y2 : V2=CS*Z2+SS*X2
330 LINE (U1,V1)-(UB,VB),KL(G) : LINE -(U2,V2),KL(G)
340 RETURN
350 M=M+1 : G=G+1 : GOSUB 240 : GOSUB 300
360 XA=XX : YA=YY : ZA=ZZ
370 XB=X1 : YB=Y1 : ZB=Z1
380 XA(M)=XX : YA(M)=YY : ZA(M)=ZZ
390 XB(M)=X2 : YB(M)=Y2 : ZB(M)=Z2 : S(M)=G
400 RETURN
410 G=G+1 : GOSUB 240 : GOSUB 300
420 XA=XA(M) : YA=YA(M) : ZA=ZA(M)
430 XB=XB(M) : YB=YB(M) : ZB=ZB(M) : M=M-1
440 GOSUB 240 : GOSUB 300
450 XA=XA(M) : YA=YA(M) : ZA=ZA(M)
460 XB=XB(M) : YB=YB(M) : ZB=ZB(M)
470 G=S(M) : M=M-1 : RETURN
480 END
```

```
10 REM ***FRACTAL PROFILE***
20 REM ***NAME:PROFILE***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 RANDOMIZE TIMER
70 WINDOW (0,-1)-(639,1)
80 P=9 : N=512 : DIM F(N)
90 H=.5 : SIGMA=.15
100 REM ***INITIALISATION***
110 F(0)=.4 : F(256)=-.4 : F(512)=.2
120 DEL=SIGMA : A=256 : B=128
130 FOR M=1 TO P
140 DEL=DEL*2^(-H)
150 FOR I=B TO N-B STEP A
160 S=(F(I+B)+F(I-B))/2
170 GOSUB 250 : F(I)=S+R*DEL
180 NEXT I
190 A=A/2 : B=B/2
200 NEXT M
210 FOR I=0 TO N : X=64+I : Y=F(I)
220 IF Y>0 THEN COL=2 ELSE COL=9
230 LINE (X,0)-(X,Y),COL : NEXT I
240 A$=INPUT$(1) : END
250 R1=2*RND-1 : R2=2*RND-1 : R3=R1*R1+R2*R2
260 IF R3>1 THEN GOTO 250
270 R=R1*SQR(-2*LOG(R3)/R3)
280 RETURN : END
```

```
10 REM ***PROJECTION OF A SURFACE***
20 REM ***NAME:PROJS***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 WINDOW (0,-1)-(639,1)
70 DEF FNA(X,Y)=SIN(7*X)*COS(9*Y)
80 M1=320 : M2=256 : M=M1+M2
90 NH=40 : NV=32
100 P=.4 : Q1=10*P/9 : Q2=8*P/9 :  Q3=.2 : REM ***
    PROJECTION COEFFICIENTS***
110 DIM S(M) : FOR L=0 TO M : S(L)=-1 : NEXT L
120 FOR J=0 TO NV : Y=J/NV
130 FOR I=0 TO M1 : X=I/M1
140 Z=FNA(X,Y) : U=288+320*X-256*Y :
    V=Q1*X+Q2*Y+Q3*Z-P
150 K=M2+I-J*(M2\NV)
160 IF V<S(K) THEN GOTO 190 ELSE S(K)=V
170 PSET (U,V),10
180 IF I=0 THEN LINE (U,V)-(U,V-Q3*Z-.3),7
190 NEXT I : NEXT J
200 FOR L=0 TO M : S(L)=-1 : NEXT L
210 FOR I=0 TO NH : X=I/NH
220 FOR J=0 TO M2 : Y=J/M2
230 Z=FNA(X,Y) : U=288+320*X-256*Y :
    V=Q1*X+Q2*Y+Q3*Z-P
240 K=M2-J+I*(M1/NH)
250 IF V<S(K) THEN GOTO 280 ELSE S(K)=V
260 PSET (U,V),12
270 IF J=0 THEN LINE (U,V)-(U,V-Q3*Z-.3),7
280 NEXT J : NEXT I : END
```

```
10 REM ***THE VAN DER WAERDEN CURVE***
20 REM ***NAME:WAERDEN***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 WINDOW (-.3,-.3)-(1.3,.9)
70 LINE (0,0)-(1,0)
80 P=10
90 H=2^(-P) : FOR K=0 TO 2^P : X=K*H
100 S=0 : FOR J=0 TO 8
110 Y=2^J*X-INT(2^J*X)
120 IF Y<.5 THEN Z=Y ELSE Z=1-Y
130 S=S+Z/2^J : NEXT J
140 IF K=0 THEN PSET (X,S) ELSE LINE-(X,S)
150 NEXT K : BEEP : A$=INPUT$(1) : END
```

```
10 REM ***PROJECTION OF MT TAKAGI***
20 REM ***NAME:TAKAGI***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 WINDOW (0,-1)-(639,1)
70 M1=640 : M2=512 : M=M1+M2
80 NH=60 : NV=48
90 P=.3 : Q1=10*P/9 : Q2=8*P/9 :  Q3=.8 : REM ***
   PROJECTION COEFFICIENTS***
100 DIM S(M) : FOR L=0 TO M : S(L)=-1 : NEXT L
110 FOR J=0 TO NV : Y=J/NV
120 FOR I=0 TO M1 : X=I/M1
130 GOSUB 300 : U=288+320*X-256*Y :
    V=Q1*X+Q2*Y+Q3*Z-P
140 K=M2+I-J*(M2\NV)
150 IF V<S(K) THEN GOTO 180 ELSE S(K)=V
160 PSET (U,V),10
170 IF I=0 THEN LINE (U,V)-(U,V-Q3*Z-.2),7
180 NEXT I : NEXT J
190 FOR L=0 TO M : S(L)=-1 : NEXT L
200 FOR I=0 TO NH : X=I/NH
210 FOR J=0 TO M2 : Y=J/M2
220 GOSUB 300 : U=288+320*X-256*Y :
    V=Q1*X+Q2*Y+Q3*Z-P
230 K=M2-J+I*(M1/NH)
240 IF V<S(K) THEN GOTO 270 ELSE S(K)=V
250 PSET (U,V),12
260 IF J=0 THEN LINE (U,V)-(U,V-Q3*Z-.2),7
270 NEXT J : NEXT I
280 LINE (32,Q2-P-.2)-(288,-P-.2),7
290 LINE -(608,Q1-P-.2),7 : END
300 Z=0
310 FOR K=0 TO 6 : X1=2^K*X-INT(2^K*X) :
    Y1=2^K*Y-INT(2^K*Y)
320 GOSUB 340 : Z=Z+F/2^K
330 NEXT K : RETURN : END
340 F1=1/2-ABS(X1-1/2) : F2=1/2-ABS(Y1-1/2)
350 IF F1<F2 THEN F=F1 ELSE F=F2
360 RETURN : END
```

```
10 REM ***FRACTAL SURFACE***
20 REM ***NAME:FRSURF***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 XM=320 : IF SCR=9 THEN YM=175 ELSE YM=240
70 A=1/2 : C=10 : M=6 : N=160 : PI=4*ATN(1)
80 CLS : FOR I=0 TO N
90 FOR J=0 TO I
100 IF INKEY$<>"" THEN END
110 X=I/N : Y=J/N
120 Z=1-X : FOR K=0 TO M
130 X1=2^K*X-INT(2^K*X) : Y1=2^K*Y-INT(2^K*Y)
140 GOSUB 230 : Z=Z+A^K*F
150 NEXT K
160 COL=1+INT(C*Z) MOD 14
170 PSET (XM+I,YM+J),COL : PSET (XM-I,YM+J),COL
180 PSET (XM-I,YM-J),COL : PSET (XM+I,YM-J),COL
190 PSET (XM+J,YM+I),COL : PSET (XM-J,YM+I),COL
200 PSET (XM-J,YM-I),COL : PSET (XM+J,YM-I),COL
210 NEXT J : NEXT I
220 A$=INPUT$(1) : END
230 IF X1<1/2 THEN F1=X1 ELSE F1=1-X1
240 IF Y1<1/2 THEN F2=Y1 ELSE F2=1-Y1
250 IF F1<F2 THEN F=F1 ELSE F=F2
260 RETURN
270 F=SIN(PI*X1)*COS(PI*Y1)
280 RETURN : END
```

Hoofdstuk 11

```
10 REM ***FILLED JULIA FRACTAL OF Z:=Z^2+C ***
20 REM ***NAME:JULFILL***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
60 IF SCR=12 THEN XM=320 : YM=240 : BH=1
70 SCREEN SCR : CLS
80 A=-.7 : B=0
90 P1=1.6 : P2=1.2 : REM ***WINDOW***
100 N1=200 : N2=INT(BH*N1*P2/P1)
110 IF B=0 THEN N0=0 ELSE N0=-N2
120 FOR I=0 TO N1 : FOR J=N0 TO N2
130 IF INKEY$<>"" THEN END
140 I1=XM+I : I2=XM-I : X=I*P1/N1
150 J1=YM-J : J2=YM+J : Y=J*P2/N2
160 FOR K=1 TO 200
170 X1=X*X-Y*Y+A : Y1=2*X*Y+B
180 S=X*X+Y*Y : S1=(X-X1)*(X-X1)+(Y-Y1)*(Y-Y1)
190 IF S>1000 THEN GOTO 250
200 IF S1<.001 THEN GOTO 230
210 X=X1 : Y=Y1
220 NEXT K
230 PSET (I1,J1) : PSET (I2,J2)
240 IF B=0 THEN PSET (I2,J1) : PSET (I1,J2)
250 NEXT J : NEXT I
260 BEEP : END
```

```
10 REM ***JULIA FRACTAL OF Z*Z+C ***
20 REM ***NAME:JUL1A***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
70 IF SCR=12 THEN XM=320 : YM=240 : BH=1
80 DIM COL(16)
90 DATA 0,1,9,4,12,2,10,1,9,4,12,2,10,1,9,4,12
100 FOR I=0 TO 16 : READ COL(I) : NEXT I
110 A=.1 : B=.3 : REM ***PARAMETERS***
120 P1=1.5 : P2=1.5 : REM ***WINDOW***
130 N1=160 : N2=INT(BH*N1*P2/P1)
140 FOR I=0 TO N1 : FOR J=-N2 TO N2
150 IF INKEY$<>"" THEN END
160 I1=XM+I : I2=XM-I : X=I*P1/N1
170 J1=YM-J : J2=YM+J : Y=J*P2/N2
180 FOR K=1 TO 200
190 X1=X*X-Y*Y+A : Y1=2*X*Y+B
200 S=X*X+Y*Y : S1=(X-X1)*(X-X1)+(Y-Y1)*(Y-Y1)
210 IF S>1000 THEN L=1+K MOD 6 : GOTO 250
220 IF S1<.0001 THEN L=1+K MOD 6 : GOTO 250
230 X=X1 : Y=Y1
240 NEXT K : L=0
250 PSET (I1,J1),COL(L) : PSET (I2,J2),COL(L)
260 NEXT J : NEXT I
270 BEEP : A$=INPUT$(1) : END
```

```
10 REM ***JULIA FRACTAL OF mu*Z+(1-mu)*Z*Z WITH
   GIVEN EIGENVALUE***
20 REM ***NAME:JUL1B***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
70 IF SCR=12 THEN XM=320 : YM=240 : BH=1
80 PI=4*ATN(1) : DIM COL(16)
90 DATA 0,1,9,4,12,2,10,1,9,4,12,2,10,1,9,4,12
100 FOR I=0 TO 16 : READ COL(I) : NEXT I
110 R=1.03 : PHI=8*PI/5 : REM ***ABSOLUTE VALUE AND
    ARGUMENT***
120 DEL=1.2 : REM ***HALF WIDTH OF WINDOW***
130 A=R*COS(PHI) : B=R*SIN(PHI)
140 DEN=2*((A-1)*(A-1)+B*B)
150 XC=(A*A+B*B-A)/DEN : YC=-B/DEN
160 N1=240 : N2=INT(BH*N1*.75)
170 FOR I=0 TO N1 : FOR J=-N2 TO N2
180 X=XC+I*DEL/N1 : Y=YC+J*DEL/N2
190 IF INKEY$<>"" THEN END
200 FOR K=1 TO 200
210 S=X*X+Y*Y
220 IF S>1000 THEN L=1+K MOD 8 : GOTO 290
230 X2=X*X-Y*Y : Y2=2*X*Y
240 X1=A*X-B*Y : Y1=A*Y+B*X
250 U=X1+(1-A)*X2+B*Y2 : V=Y1+(1-A)*Y2-B*X2
260 IF (U-X)*(U-X)+(V-Y)*(V-Y)<.0001 THEN
    L=1+K MOD 8 : GOTO 290
270 X=U : Y=V
280 NEXT K : L=0
290 IF K>5 THEN PSET (XM+I,YM-J),COL(L) :
    PSET (XM-I,YM+J),COL(L)
300 NEXT J : NEXT I : BEEP
310 A$=INPUT$(1) : END
```

```
10 REM ***EXTERIOR OF JULIA FRACTAL OF Z*Z+C IN
   BINARY CODING***
20 REM ***NAME:JUL1C***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
70 IF SCR=12 THEN XM=320 : YM=240 : BH=1
80 A=0 : B=1 : REM ***PARAMETERS***
90 P1=1.8 : P2=1.6 : REM ***WINDOW***
100 FOR I=-N1 TO N1 : FOR J=-N2 TO N2
110 FOR I=-N1 TO N1 : FOR J=-N2 TO N2
120 IF INKEY$<>"" THEN END
130 I1=XM+I : X=I*P1/N1
140 J1=YM-J : Y=J*P2/N2
150 FOR K=1 TO 100
160 S=X*X+Y*Y
170 IF S>100 AND Y>O THEN L=4 : GOTO 240
180 IF S>100 AND Y<=O THEN L=14 : GOTO 240
190 X1=X*X-Y*Y+A : Y1=2*X*Y+B
200 S1=(X-X1)*(X-X1)+(Y-Y1)*(Y-Y1)
210 IF S1<.0001 THEN GOTO 250
220 X=X1 : Y=Y1
230 NEXT K : L=0
240 PSET (I1,J1),L
250 NEXT J : NEXT I
260 A$=INPUT$(1) : END
```

```
10 REM ***JULIA SET OF Z:=Z*Z+C ***
20 REM ***MONTE CARLO METHOD***
30 REM ***NAME:JULIAMC***
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS
70 INPUT"A = ",A : INPUT"B = ",B
80 RANDOMIZE 11
90 WINDOW (-2,-1.5)-(2,1.5)
100 X=RND : Y=0 : K=1
110 WHILE K<10000 AND INKEY$=""
120 X1=(X-A)/2 : Y1=(Y-B)/2 : R=SQR(X1*X1+Y1*Y1)
130 X=SQR(R+X1) : Y=SQR(R-X1)
140 IF Y1<0 THEN Y=-Y
150 IF RND<.5 THEN X=-X : Y=-Y
160 IF K>32 THEN GOSUB 190
170 K=K+1 : WEND
180 A$=INPUT$(1) : END
190 PSET (X,Y) : PSET (-X,-Y)
200 IF B=0 THEN PSET (-X,Y) : PSET (X,-Y)
210 RETURN : END
```

```
10 REM ***JULIA SET OF Z*Z+C WITH COMPLEX C ***
20 REM ***BACKTRACKING WITH REDUCTION BY AREA
   SATURATION***
30 REM ***NAME:JULIABT***
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS
70 A=-.12 : B=.74 : P=64
80 DIM X(P),Y(P),Z(P)
90 L=2 : H=1.5 : WINDOW (-L,-H)-(L,H)
100 M1=160 : M2=120 : DIM T%(160,120)
110 D1=2*L/M1 : D2=2*H/M2 : TMAX=8
120 A1=1/8-A/2 : B1=-B/2 : R1=SQR(A1*A1+B1*B1)
130 XD=1/2+SQR(R1+A1) : YD=SQR(R1-A1)
140 IF B>0 THEN YD=-YD : REM ***REPELLING FIXED
    POINT***
150 X=-XD : Y=-YD : M=0 : G=0
160 N1=INT((L-X)/D1) : N2=INT((H-Y)/D2)
170 IF T%(N1,N2)<TMAX THEN PSET (X,Y) : PSET (-X,-Y)
180 T%(N1,N2)=T%(N1,N2)+1
190 IF INKEY$<> "" THEN GOTO 310
200 IF G<P AND T%(N1,N2)<2*TMAX THEN GOSUB 220
    ELSE GOSUB 290
210 IF M>=0 THEN GOTO 160 ELSE GOTO 310
220 X0=(X-A)/2 : Y0=(Y-B)/2
230 R=SQR(X0*X0+Y0*Y0)
240 X1=-SQR(R+X0) : Y1=-SQR(R-X0)
250 IF Y0<0 THEN Y1=-Y1
260 X2=-X1 : Y2=-Y1
270 M=M+1 : G=G+1
280 X=X1 : Y=Y1 : X(M)=X2 : Y(M)=Y2 : Z(M)=G : RETURN
290 X=X(M) : Y=Y(M) : G=Z(M) : M=M-1
300 RETURN
310 BEEP : A$=INPUT$(1) : END
```

Hoofdstuk 12

```
10 REM ***JULIA SET WITH DISTANCE FORMULA***
20 REM ***NAME:JULDIST***
30 DEFDBL A,B,D,S,X,Y,U,V
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS
70 IF SCR=9 THEN XM=320 : YM=175 : HB=35/48
80 IF SCR=12 THEN XM=320 : YM=240 : HB=1
90 A=.11031 : B=.67037 : REM ***C=A+Bi***
100 DELH=1.4 : DELV=1.4
110 N1=160 : N2=INT(HB*N1*DELV/DELH)
120 FOR I=0 TO N1 : FOR J=-N2 TO N2
130 I1=XM+I : I2=XM-I : J1=YM-J : J2=YM+J
140 IF I=0 AND J=0 THEN GOTO 250
150 IF INKEY$<>"" THEN END
160 X=I*DELH/N1 : Y=J*DELV/N2
170 U=1 : V=0
180 FOR K=0 TO 400
190 X1=X*X-Y*Y+A : Y1=2*X*Y+B
200 U1=2*(U*X-V*Y) : V1=2*(U*Y+V*X)
210 S1=X1*X1+Y1*Y1+1E-10
220 IF S1>256 THEN GOTO 260
230 X=X1 : Y=Y1 : U=U1 : V=V1
240 NEXT K
250 NEXT J : NEXT I : END
260 S2=LOG(S1) : S3=SQR(U1*U1+V1*V1)
270 S4=EXP(S2/2^K)
280 DIST=SQR(S1)*S2/(S3*S4)
290 IF DIST<.5*DELH/N1 THEN GOSUB 310 ELSE GOSUB 320
300 GOTO 250
310 PSET (I1,J1),14 : PSET (I2,J2),14 : RETURN
320 PSET (I1,J1),1 : PSET (I2,J2),1 : RETURN
330 END
```

```
10 REM ***JULIA FRACTAL OF Z(C+(1-C)Zˆ6) ***
20 REM ***NAME:JULROT1***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN =",SCR
50 DEFDBL A,B,C,D,P,R,X,Y,Z
60 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
70 IF SCR=12 THEN XM=320 : YM=240 : BH=1
80 SCREEN SCR : CLS
90 DIM COL(16)
100 DATA 0,1,9,2,10,4,12,6,1,9,2,10,4,12,1,9,2
110 FOR I=0 TO 16 : READ COL(I) : NEXT I
120 P=1.5 : REM ***CIRCULAR WINDOW***
130 A=1.5 : B=0 : REM ***PARAMETERS***
140 C1=1000 : C2=.0001 : REM ***ESCAPE BOUNDS***
150 N1=160 : N2=INT(BH*N1)
160 FOR I=0 TO N1 : FOR J=0 TO N2
170 IF INKEY$<>"" THEN END
180 X=P*I/N1 : Y=P*J/N2
190 IF X*X+Y*Y>P*P THEN GOTO 310
200 FOR K=1 TO 40
210 X2=X*X-Y*Y : Y2=2*X*Y
220 X4=X2*X2-Y2*Y2 : Y4=2*X2*Y2
230 X6=X2*X4-Y2*Y4 : Y6=X2*Y4+Y2*X4
240 D1=X6*X6+Y6*Y6 : D2=(X6-1)*(X6-1)+Y6*Y6
250 IF D1>C1 OR D2<C2 THEN L=1+K MOD 8 : GOTO 290
260 X0=A*(1-X6)+B*Y6+X6 : Y0=B*(1-X6)-A*Y6+Y6
270 Z=X : X=X*X0-Y*Y0 : Y=Z*Y0+Y*X0
280 NEXT K : L=0
290 PSET (XM+I,YM-J),COL(L) : PSET (XM-I,YM+J),COL(L)
300 PSET (XM+I,YM+J),COL(L) : PSET (XM-I,YM-J),COL(L)
310 NEXT J : NEXT I
320 A$=INPUT$(1) : END
```

```
10 REM ***JULIA FRACTAL OF Z(C+(1-C)/Zˆ6) ***
20 REM ***NAME:JULROT2***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 DEFDBL A,B,C,D,P,R,X,Y,Z
60 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
70 IF SCR=12 THEN XM=320 : YM=240 : BH=1
80 SCREEN SCR : CLS
90 DIM COL(16)
100 DATA 0,1,9,2,10,4,12,6,1,9,2,10,4,12,1,9,2
110 FOR I=0 TO 16 : READ COL(I) : NEXT I
120 P=20 : REM ***CIRCULAR WINDOW***
130 A=.8 : B=.05 : REM ***PARAMETERS***
140 N1=200 : N2=INT(BH*N1)
150 FOR I=0 TO N1 : FOR J=-N2 TO N2
160 IF INKEY$<>"" THEN END
170 X=P*I/N1 : Y=P*J/N2
180 IF X*X+Y*Y>P*P THEN GOTO 310
190 FOR K=1 TO 40
200 X2=X*X-Y*Y : Y2=2*X*Y
210 X4=X2*X2-Y2*Y2 : Y4=2*X2*Y2
220 X6=X2*X4-Y2*Y4 : Y6=X2*Y4+Y2*X4
230 D=(X6-1)*(X6-1)+Y6*Y6
240 IF D<.0001 THEN L=1+K MOD 8 : GOTO 300
250 DEN=X6*X6+Y6*Y6+.00001
260 XNUM=X6/DEN : YNUM=-Y6/DEN
270 X0=A*(1-XNUM)+B*YNUM+XNUM :
    Y0=B*(1-XNUM)-A*YNUM+YNUM
280 Z=X : X=X*X0-Y*Y0 : Y=Z*Y0+Y*X0
290 NEXT K : L=0
300 PSET (XM+I,YM-J),COL(L) :
    PSET (XM-I,YM+J),COL(L)
310 NEXT J : NEXT I
320 A$=INPUT$(1) : END
```

```
10 REM ***JULIA FRACTAL OF C+Zˆ3 WITH SPECIAL
   BOUNDARY***
20 REM ***NAME:JULCELL***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
70 IF SCR=12 THEN XM=320 : YM=240 : BH=1
80 DIM COL(16)
90 DATA 0,1,9,4,12,6,14,5,13,1,9,4,12,6,14,5,13
100 FOR I=0 TO 16 : READ COL(I) : NEXT I
110 P=1.8 : REM ***RADIUS CIRCULAR WINDOW***
120 A=-.6525 : B=.34 : D=.2 : REM ***PARAMETER
    VALUES***
130 N1=160 : N2=INT(BH*N1)
140 FOR I=-N1 TO N1 : FOR J=-N2 TO N2
150 X=I*P/N1 : Y=J*P/N2 : K=0
160 IF X*X+Y*Y>P*P THEN GOTO 260
170 IF INKEY$<>"" THEN END
180 FOR K=1 TO 40
190 T1=ABS(Y)+D*ABS(X) : T2=ABS(X)+D*ABS(Y)
200 IF T1>20 AND T2>20 THEN GOTO 240
210 X3=X*(X*X-3*Y*Y) : Y3=Y*(3*X*X-Y*Y)
220 X=X3+A : Y=Y3+B
230 NEXT K : GOTO 260
240 L=1+(K+1) MOD 16
250 PSET (XM+I,YM-J),COL(L)
260 NEXT J : NEXT I
270 A$=INPUT$(1) : END
```

```
10 REM ***JULIA FRACTAL OF Z:=Zˆ3-3Aˆ2 Z+B ***
20 REM ***NAME:JULCUB***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
70 IF SCR=12 THEN XM=320 : YM=240 : BH=1
80 DIM COL(8) : DATA 0,1,9,2,10,14,6,4,12
90 FOR I=0 TO 8 : READ COL(I) : NEXT I
100 AR=-.57695 : AI=.175 : BR=.8 : BI=0
110 A1=3*(AR*AR-AI*AI) : A2=6*AR*AI : B1=BR : B2=BI
120 XC=-.14 : YC=0 : REM ***CENTRE WINDOW***
130 DELH=1.7 : DELV=1.2 : REM ***DIMENSIONS WINDOW***
140 N1=160 : N2=INT(BH*N1*DELV/DELH)
150 FOR I=-N1 TO N1 : FOR J=-N2 TO N2
160 IF INKEY$<>"" THEN END
170 IF INKEY$<>"" THEN END
180 X=XC+I*DELH/N1 : Y=YC+J*DELV/N2
190 FOR K=0 TO 100
200 S=X*X+Y*Y
210 IF S>1000 THEN L=1+K MOD 8 : GOTO 270
220 X3=X*(X*X-3*Y*Y)-A1*X+A2*Y+B1
230 Y3=Y*(3*X*X-Y*Y)-A1*Y-A2*X+B2
240 IF (X-X3)*(X-X3)+(Y-Y3)*(Y-Y3)<.0001
    THEN L=1+K MOD 8 : GOTO 270
250 X=X3 : Y=Y3
260 NEXT K : L=0
270 PSET (XM+I,YM-J),COL(L)
280 NEXT J : NEXT I
290 BEEP : A$=INPUT$(1) : END
```

```
10 REM ***JULIA SET OF EXP(Z)-Z-1 ***
20 REM ***NAME:JULEXP***
30 PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
70 IF SCR=12 THEN XM=320 : YM=240 : BH=1
80 DIM COL(8) : DATA 0,1,9,2,10,4,12,14,6
90 FOR I=0 TO 8 : READ COL(I) : NEXT I
100 DELH=4 : DELV=5
110 N1=120 : N2=INT(BH*N1*DELV/DELH)
120 FOR I=-N1 TO N1
130 FOR J=0 TO N2
140 IF INKEY$<>"" THEN END
150 X=I*DELH/N1 : Y=J*DELV/N2
160 FOR K=0 TO 40
170 IF X>6 THEN L=1+K MOD 8 : GOTO 220
180 X1=EXP(X)*COS(Y)-X-1 : Y1=EXP(X)*SIN(Y)-Y
190 IF (X-X1)*(X-X1)+(Y=Y1)*(Y-Y1)<.000001
    THEN L=1+K MOD 8 : GOTO 220
200 X=X1 : Y=Y1
210 NEXT K : GOTO 230
220 PSET(XM+I,YM-J),COL(L) : PSET (XM+I,YM+J),COL(L)
230 NEXT J : NEXT I : BEEP
240 A$=INPUT$(1) : END
```

```
10 REM ***JULIA SET OF Z:=C SIN Z ***
20 REM ***NAME:JULSIN***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS : PI=4*ATN(1)
60 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
70 IF SCR=12 THEN XM=320 : YM=240 : BH=1
80 DIM COL(8) : DATA 0,1,9,2,10,4,12,14,6
90 FOR I=0 TO 8 : READ COL(I) : NEXT I
100 A=.6 : B=.6
110 DELH=1.5*PI : DELV=4
120 N1=200 : N2=INT(BH*N1*DELV/DELH)
130 FOR I=0 TO N1
140 FOR J=-N2 TO N2
150 IF INKEY$<>"" THEN END
160 X=I*DELH/N1 : Y=J*DELV/N2
170 FOR K=0 TO 50
180 IF ABS(Y)>20 THEN L=1+K MOD 8 : GOTO 270
190 U=EXP(Y) : V=EXP(-Y)
200 CH=(U+V)/2 : SH=(U-V)/2
210 CS=COS(X) : SS=SIN(X)
220 X1=SS*CH : Y1=CS*SH
230 X2=A*X1-B*Y1 : Y2=A*Y1+B*X1
240 IF (X2-X)*(X2-X)+(Y2-Y)*(Y2-Y)<.000001
    THEN L=1+K MOD 8 : GOTO 270
250 X=X2 : Y=Y2
260 NEXT K : GOTO 280
270 PSET (XM+I,YM-J),COL(L) : PSET (XM-I,YM+J),COL(L)
280 NEXT J : NEXT I : BEEP
290 A$=INPUT$(1) : END
```

Hoofdstuk 13

```
10 REM ***MANDELBROT SET, TOTAL VIEW***
20 REM ***NAME:MANTOT***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 DELH=1.7 : DELV=1.5
70 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
80 IF SCR=12 THEN XM=320 : YM=240 : BH=1
90 N1=240 : N2=(BH*N1*DELV/DELH)
100 DIM COL(8) : DATA 0,1,9,2,10,4,12,6,14
110 FOR I=0 TO 8 : READ COL(I) : NEXT I
120 FOR I=-N1 TO N1 : A=-.65+I*DELH/N1
130 FOR J=0 TO N2 : B=J*DELV/N2
140 IF INKEY$<>"" THEN END
150 U=4*(A*A+B*B) : V=U-2*A+1/4
160 IF U+8*A+15/4<0 THEN L=0 : GOTO 240
170 IF V-SQR(V)+2*A-1/2<0 THEN L=0 : GOTO 240
180 X=A : Y=B
190 FOR K=1 TO 100
200 U=X*X : V=Y*Y : W=2*X*Y
210 X=U-V+A : Y=W+B
220 IF U+V>16 THEN L=1+K MOD 8  : GOTO 240
230 NEXT K : L=0
240 PSET (XM+I,YM+J),COL(L)
250 PSET (XM+I,YM-J),COL(L)
260 NEXT J : NEXT I : BEEP
270 A$=INPUT$(1) : END
```

```
10 REM ***MANDELBROT SET, DETAIL***
20 REM ***NAME:MANDET***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 INPUT"A = ",AC : INPUT"B = ",BC
70 INPUT"DEL = ",DEL : INPUT"NMAX =",NMAX
80 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
90 IF SCR=12 THEN XM=320 : YM=240 : BH=1
100 N1=200 : N2=(.75*BH*N1)
110 IF B=0 THEN N0=0 ELSE N0=-N2
120 DIM COL(8) : DATA 0,1,9,2,10,4,12,6,14
130 FOR I=0 TO 8 : READ COL(I) : NEXT I
140 FOR I=-N1 TO N1 : A=AC+I*DEL/N1
150 FOR J=N0 TO N2 : B=BC+J*.75*DEL/N2
160 IF INKEY$<>"" THEN END
170 X=A : Y=B
180 FOR K=1 TO NMAX
190 U=X*X : V=Y*Y : W=2*X*Y
200 X=U-V+A : Y=W+B
210 IF U+V>16 THEN L=1+K MOD 8  : GOTO 230
220 NEXT K : L=0
230 PSET (XM+I,YM-J),COL(L)
240 IF B=0 THEN PSET (XM+I,YM+J),COL(L)
250 NEXT J : NEXT I : BEEP
260 A$=INPUT$(1) : END
```

```
10 REM ***MANDELBROT SET WITH DISTANCE FORMULA***
20 REM ***NAME:MANDIST***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN NUMBER = ",NSCR
50 SCREEN NSCR : CLS
60 DEFDBL A,B,D,S,X,Y,U,V
70 XM=320 : IF NSCR=9 THEN YM=175 : HB= 35/48 ELSE
   YM=240 : HB=1
80 AC=-.16 : BC=1.03 : REM ***CENTRE WINDOW***
90 DEL=.025 : REM ***HALVE HORIZONTAL WIDTH***
100 N1=200 : N2=INT(.75*HB*N1)
110 FOR I=-N1 TO N1 : FOR J=-N2 TO N2
120 IF INKEY$<>"" THEN END
130 A=AC+I*DEL/N1 : B=BC+J*DEL/N1
140 X=A : Y=B : U=1 : V=0
150 FOR K=0 TO 200
160 X1=X*X-Y*Y+A : Y1=2*X*Y+B
170 U1=2*(U*X-V*Y)+1 : V1=2*(U*Y+V*X)
180 S1=X1*X1+Y1*Y1
190 IF S1>64 THEN GOTO 230
200 X=X1 : Y=Y1 : U=U1 : V=V1
210 NEXT K
220 NEXT J : NEXT I : END
230 S2=LOG(S1) : S3=SQR(U1*U1+V1*V1)
240 DIST=SQR(S1)*S2/S3
250 IF DIST<.4*DEL/N1 THEN PSET (XM+I,YM-J),14
260 GOTO 220
270 END
```

```
10 REM ***MANDELBROT SET OF C+Z-EXP(Z) ***
20 REM ***NAME:MANEXP***
30 DEFDBL A,B,D,X,Y,U
40 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
50 PRINT : INPUT"SCREEN = ",SCR
60 SCREEN SCR : CLS
70 XM=320 : IF SCR=9 THEN YM=175 : HB=35/48
   ELSE YM=240 : HB=1
80 DIM COL(8) : DATA 0,1,9,2,10,6,14,4,12
90 FOR I=0 TO 8 : READ COL(I) : NEXT I
100 AC=4 : BC=0 : DELH=4 : DELV=3.15
110 N1=200 : N2=INT(HB*N1*DELV/DELH)
120 FOR I=-N1 TO N1 : A=AC+I*DELH/N1
130 FOR J=0 TO N2 : B=BC+J*DELV/N2
140 IF INKEY$<>"" THEN END
150 IF (A-1)*(A-1)+B*B<1 THEN GOTO 250
160 X=0 : Y=0
170 FOR K=1 TO 20
180 IF X>10 THEN L=1+K MOD 8 : GOTO 240
190 IF X<-10 THEN U=0 ELSE U=EXP(X)
200 X1=A+X-U*COS(Y) : Y1=B+Y-U*SIN(Y)
210 IF (X-X1)*(X-X1)+(Y-Y1)*(Y-Y1)<.0001
    THEN GOTO 250
220 X=X1 : Y=Y1
230 NEXT K : GOTO 250
240 PSET (XM+I,YM-J),COL(L) : PSET (XM+I,YM+J),COL(L)
250 NEXT J : NEXT I
260 BEEP : END
```

```
10 REM ***MANDELBROT SET OF C+Z-SIN(Z) ,TOTAL VIEW***
20 REM ***NAME:MANSIN1***
30 CLS : PRINT"PRESS9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
70 IF SCR=12 THEN XM=320 : YM=240 : BH=1
80 DIM COL(8) : PI=4*ATN(1)
90 DATA 0,3,11,1,9,2,10,14,6
100 FOR I=0 TO 8 : READ COL(I) : NEXT I
110 DELH=2*PI : DELV=1.5*PI
120 N1=200 : N2=INT(BH*N1*DELV/DELH)
130 FOR I=0 TO N1 : I1=XM+I : I2=XM-I
140 FOR J=0 TO N2 : J1=YM+J : J2=YM-J
150 IF INKEY$<>"" THEN END
160 A=DELH*I/N1 : B=DELV*J/N2
170 SC=A*A+B*B : C=(5-SC*SC)/4
180 D=1+2*C-2*C*C
190 IF A*A<B*B+D THEN L=3 : GOTO 300
200 X=0 : Y=0
210 FOR K=1 TO 100
220 IF ABS(Y)>12 THEN GOTO 290
230 U=EXP(Y) : V=1/U : CH=(U+V)/2 : SH=(U-V)/2
240 SS=SIN(X) : CS=COS(X)
250 X1=A+X-SS*CH : Y1=B+Y-CS*SH
260 IF ABS(X-X1)+ABS(Y-Y1)<.001 THEN L=3 : GOTO 300
270 X=X1 : Y=Y1
280 NEXT K : L=3 : GOTO 300
290 IF K<50 THEN L=1+K MOD 8 ELSE L=3
300 PSET (I1,J1),COL(L) : PSET (I1,J2),COL(L)
310 PSET (I2,J1),COL(L) : PSET (I2,J2),COL(L)
320 NEXT J : NEXT I : BEEP
330 A$=INPUT$(1) : END
```

```
10 REM ***MANDELBROT SET OF C+PI*SIN(Z)
   TOTAL VIEW***
20 REM ***NAME:MANSIN2***
30 CLS : PRINT"PRESS 9 FOR EGA OR 12 FOR VGA"
40 DEFDBL A,B,C,D,S,U,V,X,Y
50 PRINT : INPUT"SCREEN = ",SCR
60 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
70 IF SCR=12 THEN XM=320 : YM=240 : BH=1
80 SCREEN SCR : CLS
90 DIM COL(8) : DP=4*ATN(1)
100 DATA 0,1,9,2,10,4,12,14,6
110 FOR I=0 TO 8 : READ COL(I) : NEXT I
120 XM=320 : YM=240 : N1=200 : N2=INT(.75*BH*N1)
130 AC=0 : BC=0 : DEL=DP : REM ***CENTRE WINDOW
    AND HALF WIDTH***
140 FOR I=0 TO N1 : A=AC+I*DEL/N1
150 FOR J=0 TO N2 : B=BC+J*DEL/N1
160 IF INKEY$<>"" THEN END
170 X=DP/2 : Y=0 : FOR K=1 TO 40
180 IF ABS(Y)>12 THEN GOTO 250
190 U=EXP(Y) : V=1/U : CH=(U+V)/2 : SH=(U-V)/2
200 SS=SIN(X) : CS=COS(X)
210 X1=A+DP*SS*CH : Y1=B+DP*CS*SH
220 IF ABS(X-X1)+ABS(Y-Y1)<.001 THEN GOTO 280
230 X=X1 : Y=Y1
240 NEXT K : GOTO 280
250 L=1+K MOD 8 : C=COL(L)
260 PSET (XM+I,YM-J),C : PSET (XM+I,YM+J),C
270 PSET (XM-I,YM-J),C : PSET (XM-I,YM+J),C
280 NEXT J : NEXT I : BEEP
290 A$=INPUT$(1) : END
```

```
10 REM ***PSEUDO MANDELBROT SET FOR NON CRITICAL
   ORBIT***
20 REM ***NAME:MANVAR***
30 CLS : PRINT"PRESS  9 FOR EGA OR 12 FOR VGA"
40 PRINT : INPUT"SCREEN = ",SCR
50 SCREEN SCR : CLS
60 IF SCR=9 THEN XM=320 : YM=175 : BH=35/48
70 IF SCR=12 THEN XM=320 : YM=240 : BH=1
80 DIM COL(8) : DATA 0,1,9,2,10,4,12,14,8
90 FOR I=0 TO 8 : READ COL(I) : NEXT I
100 AC=-.2 : BC=0 : DELH=1.8 : DELV=1.5
110 N1=200 : N2=INT(BH*N1*DELV/DELH)
120 FOR I=-N1 TO N1
130 A=AC+I*DELH/N1
140 FOR J=0 TO N2
150 B=BC+J*DELV/N2
160 IF INKEY$<>"" THEN END
170 X=0 : Y=.6 : REM ***START FROM A NON CRITICAL
    POINT***
180 FOR K=1 TO 50
190 Z=X : X=X*X-Y*Y+A : Y=2*Z*Y+B
200 S=X*X+Y*Y
210 IF S>100 THEN GOTO 230
220 NEXT K : GOTO 250
230 L=1+K MOD 8
240 IF K>2 THEN PSET (XM+I,YM-J),COL(L) :
    PSET (XM+I,YM+J),COL(L)
250 NEXT J : NEXT I : BEEP
260 A$=INPUT$(1) : END
```

Aharony, Amnon en Jens Feder, eds., *Fractals in Physics. Essays in Honour of Benoit B. Mandelbrot*. North-Holland Amsterdam, Physica D Vol. 38 (blz. 398), 1989

Algorithm. The personal programming newsletter. Ed. A. K. Dewdney, 2 Oxford Street West, Londen, Ontario N6H 1P9, Canada, 1990

Aono Masaki en Tosiyasu L. Kunii, 'Botanical tree image generation.' In: IEEE *Computer Graphics and Applications* 20,4, 1989

Barnsley, Michael F., *Fractals Everywhere*. Academic Press (blz. 283), 1988

Barnsley, Michael F. en Stephen G. Demko, eds., *Chaotic Dynamics and Fractals*. Academic Press (blz. 292), 1986

Blanchard, Paul, 'Complex analytical dynamics on the Riemann sphere.' In: *Bulletin of the American Mathematical Society* 11,1, 1984

Broer, H. W. en F. Verhulst, eds., *Dynamische Systemen en Chaos*. Epsilon Uitgaven nr. 14, 1990

Cellular Automata. Proceedings of an Interdisciplinary Workshop. Los Alamos, March 1983. North-Holland Amsterdam, Physica D Vol. 10, 1 & 2 (blz. 247), 1984

Devaney, Robert L., *An Introduction to Chaotic Dynamical Systems*. The Benjamin/Cummings Publishing Company (blz. 320), 1986

Devaney, Robert L. en Linda Keen, eds., *Chaos and Fractals. The Mathematics behind the Computer Graphics*. Vol. 9, Proceedings of symposia in applied mathematics, American Mathematical Society (blz. 148), 1989

Estvanik, Steve, 'From fractals to graftals.' In: *Computer Languages* 3,7, 1986

Falconer, Kenneth, *Fractal Geometry, Mathematical Foundations and Applications*. John Wiley & Sons (blz. 288), 1990

Feder, Jens, *Fractals*. Plenum Press (blz. 394), 1988

GEO Wissen, *Chaos + Kreativität*. Nr. 2, Hamburg, 1990

Lauwerier, Hans A., *Fractals. Meetkundige Figuren in Eindeloze Herhaling*. Aramith (blz. 168), 1987

Peitgen, H.-O. en P. H. Richter, *The Beauty of Fractals. Images of Complex Dynamical Systems*. Springer-Verlag (blz. 199), 1986

Peitgen Heinz-Otto en Dietmar Saupe, eds., *The Science of Fractal Images*. Springer-Verlag (blz. 312), 1988

Pickover, Clifford A., *Computers, Pattern Chaos and Beauty*. St. Martin's Press, New York, 1990

Smith, Alvyn Ray, 'Plants, fractals and formal languages.' In: *Computer Graphics* 18,3, 1984

Stanley, H. E. en Ostrowsky, N., eds., *On Growth and Form: Fractal and Non-fractal Patterns in Physics*. Martinus Nijhoff Publishers, 1986

Francis Crick, **Geniaal gekkenwerk** f 39,50
De bestseller autobiografie van de ontdekker van het DNA

Felix Eijgenraam, **In dienst van de verwondering** f 14,90
Interviews met wetenschappers

Bruno Ernst, **Avonturen met onmogelijke figuren** f 22,50

Bruno Ernst, **Nieuwe avonturen met onmogelijke figuren** f 29,90
Het verhaal over de ONdingen van Escher, Reutersvård e.a.

Nicholas Falletta, **Logische en onlogische paradoxen** f 34,90
Ze blijven boeien, vooral als blijkt dat sommige paradoxen echt 'kunnen'

Richard Feynman, **Heel geestig, meneer Feynman** f 39,50

Richard Feynman, **Laat ze maar praten** f 39,50
De autobiografie van het natuurkundegenie uit de VS, in twee delen

Richard Feynman, **QED** f 34,90
De quantum elektrodynamica kan alleen begrepen worden als Feynman het uitlegt

Flood en Lockwood, **Onomkeerbaarheid van de tijd** f 34,90
Is tijd asymmetrisch of kan ze in twee richtingen bewegen

Martin Gardner, **Spiegelsymmetrie** f 32,50
Niet Gardners gemakkelijkste, wel zijn beroemdste boek

J.E. Gordon, **Krachten in evenwicht** f 39,50
Waarom dingen niet omvallen. Bouwkunde voor leken

Hans Lauwerier, **Symmetrie. Regelmatige structuren in de kunst** f 34,90

Hans Lauwerier, **Oneindigheid, een onbereikbaar ideaal** f 32,50
De bestseller-auteur van de Fractals vertelt verder

Peter Sprent, **Risico's** f 39,50
Over onzekerheden bestaat heel wat zekerheid

Henk Tennekes, **Dan leef ik liever in onzekerheid** f 29,90
Een Nederlandse wetenschapper aan het woord

Henk Tennekes e.a., **De vlinder van Lorenz** f 34,90
Een Nederlands standaardwerk over chaos

Fractals hebben zich reeds een bestaansrecht verworven in onze wereld. Wie ze eenmaal heeft leren kennen komt ze overal tegen: een vogelveer en een varenblad, wolken in de lucht en golven in de zee, de takken van een boom en de bloedvaten in de lever, allemaal fractals.

Fractals, grillige meetkundige figuren met een soort zelfgelijkvormigheid, waarbij een willekeurig klein detail weer alle kenmerken van het geheel lijkt te hebben. Mandelbrot heeft ons geleerd ze te karakteriseren, met een zogenaamde gebroken (fractal) dimensie. Zijn ideeën hebben een stormachtige ontwikkeling ingeluid en de laatste jaren zijn er op het gebied van theorie en toepassingen spectaculaire vorderingen gemaakt. Reeds hebben ze geleid tot een nieuwe manier om de enorme hoeveelheid fotometrische gegevens, verzameld door een kunstmatige satelliet, op drastische wijze te reduceren.

Barnsley, aan wie we deze toepassing te danken hebben, laat zien hoe fractals als het ware op bestelling geleverd kunnen worden, en hoe bij voorbeeld een varenblad op eenvoudige wijze als fractal op een computerscherm zichtbaar gemaakt kan worden. Het desbetreffende programma, en nog vele andere, zal men in dit boek aantreffen. De moderne computertechnologie stelt ons nu in staat zelf talloze fractals in vele kleuren te ontwerpen.

Nodig is slechts een goede computer, wat fantasie, en een klein beetje wiskunde. Fractals kunnen gelden als objecten van computerkunst, en met behulp van wat in dit boek besproken wordt kan men, achter de computer zittend, zich een kunstenaar wanen.

Hans Lauwerier, auteur van FRACTALS, MEETKUNDIGE FIGUREN IN EINDELOZE HERHALING, is Nederlands fractal-deskundige bij uitstek. Lauwerier heeft sinds de publikatie van zijn boek een nieuwe visie ontwikkeld op het fenomeen fractals. Daaruit zijn nieuwe mogelijkheden ontstaan met indrukwekkende computergraphics.

ISBN 90 6834 076 X

ARAMITH UITGEVERS

Lauwerier

Een wereld van Fractals